TOPICS IN GENOMIC IMAGE PROCESSING

A Dissertation

by

JIANPING HUA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2004

Major Subject: Electrical Engineering

TOPICS IN GENOMIC IMAGE PROCESSING

A Dissertation

by

JIANPING HUA

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Zixiang Xiong (Chair of Committee)	Andrew K. Chan (Member)
Edward R. Dougherty (Member)	Costas N. Georghiades (Member)
Andreas Klappenecker (Member)	Chanan Singh (Head of Department)

December 2004

Major Subject: Electrical Engineering

ABSTRACT

Topics in Genomic Image Processing. (December 2004)
Jianping Hua, B.E., Tsinghua University, P.R. China;
M.S., Tsinghua University, P.R. China
Chair of Advisory Committee: Dr. Zixiang Xiong

The image processing methodologies that have been actively studied and developed now play a very significant role in the flourishing biotechnology research. This work studies, develops and implements several image processing techniques for M-FISH and cDNA microarray images. In particular, we focus on three important areas: M-FISH image compression, microarray image processing and expression-based classification. Two schemes, embedded M-FISH image coding (EMIC) and Microarray BASICA: Background Adjustment, Segmentation, Image Compression and Analysis, have been introduced for M-FISH image compression and microarray image processing, respectively. In the expression-based classification area, we investigate the relationship between optimal number of features and sample size, either analytically or through simulation, for various classifiers.

To my wife and my parents

ACKNOWLEDGMENTS

First of all I would like to thank my advisor, Professor Zixiang Xiong, for his steadfast support, constant encouragement and expert guidance in my research. He has provided me an environment conducive to learning and quality research. I have equally deep appreciation to Professor Edward R. Dougherty, for his optimism, extensive knowledge and deep insights. I would also like to thank Professor Andrew K. Chan, Professor Costas N. Georghiades and Professor Andreas Klappenecker for serving as my committee members. I am much indebted to Dr. Qiang Wu who has been of great help during my internship. I would also like to thank Dr. Yidong Chen for the creative talks we had.

Furthermore, I owe my appreciation to the colleagues in the multimedia lab, genomic signal processing lab, wireless lab and Advanced Digital Imaging Research. It is a pleasure to having worked with all of you. In particular I would like to thank Zhongmin Liu, Samuel Cheng, Jianhong Jiang, Tianli Chu, Yong Sun, Zhixin Liu, Yang Yang, Qian Xu, Min Dai, Vladimir Stankovic, Xiaobo Zhou, Yuefei Xiao, Chao Sima, Ashish Choudhary, Ranadip Pal, Sanju Nair, Ivan Ivanov, Ulisses Braga-Neto, Yoganand Balagurunathan, Jun Zheng, Shengjie Zhao, Jing Li, Yu Zhang, Wenyan He, Beng Lu, Zigang Yang, Yan Wang, Xianyou Li, Yu-ping Wang and Tiehan Chen for their sincere help. In addition, I want to thank my friends back in China: Zhiwei You, Wei Wang, Yiduo Yu, Liang Zhang, Pingping Zhuang, Deng Lu, and many many others for your encouragement and friendship.

I would like to express my profound gratitude to my parents and my loving wife, Shaoyan Zhang for their love and support.

Finally, I would like to thank everyone else whose names I have not yet mentioned. Without your generous help and support, I would never have finished this dissertation.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

Since the introduction of the first transgenic plants in 1983, the modern biotechnology has bloomed into a $200 billion industry, extending from pure scientific research into daily merchandises such as food and medicine[1]. In spite of the fiery debate in the ethics aspect, the modern biotechnology exhibits substantial importance to medical research. Nowadays, more than 4000 medical disorders caused by defective genes have been identified, and one out of ten people encounters at least one type of such disorders in his/her lifetime. These facts raise intensive demands in the development of biotechnology in various areas. In treatment, the first case of gene therapy took place in 1990 at NIH. In diagnosis, it is predicted that genetic tests on 25 diseases will be available in most hospitals in 10 years, including commonly seen diseases such as cancer and diabetes. In pharmaceutics, Gleevec, a promising new drug for leukemia, has been put into market. It along with several other drugs reveal the trend of a new generation of medicines designed under the principles of a new science subject named *pharmacogenomics*. The fast developing technology now exceeds far beyond biology itself, and poses challenging problems in various areas. Due to the multidisciplinary nature of genome-related research, researchers of different backgrounds have been summoned to contribute to this promising field.

Among all these cross-over areas, the methodologies that have been studied and developed by the image processing community – in particular, image processing, compression, signal estimation and pattern recognition, are among the most powerful tools for biologists and medical doctors. In modern biotechnology, a huge amount of data

---

The journal model is *IEEE Transactions on Automatic Control*.

are now obtained in image format, hence raise extraordinary demands on efficient genomic image processing and related data/signal processing. For example, some images for direct inspections by the physicians require highly efficient compression and transmission, and some images for further analysis necessitate accurate information extraction. Also the data obtained through image processing call for powerful signal processing and data mining technology to help biologists understand the true biological meanings behind them. The work proposed here is intended to deal with some of the most important image processing issues associated with two types of genomic image: multiplex fluorescence *in situ* hybridization (M-FISH) image and cDNA microarray image.

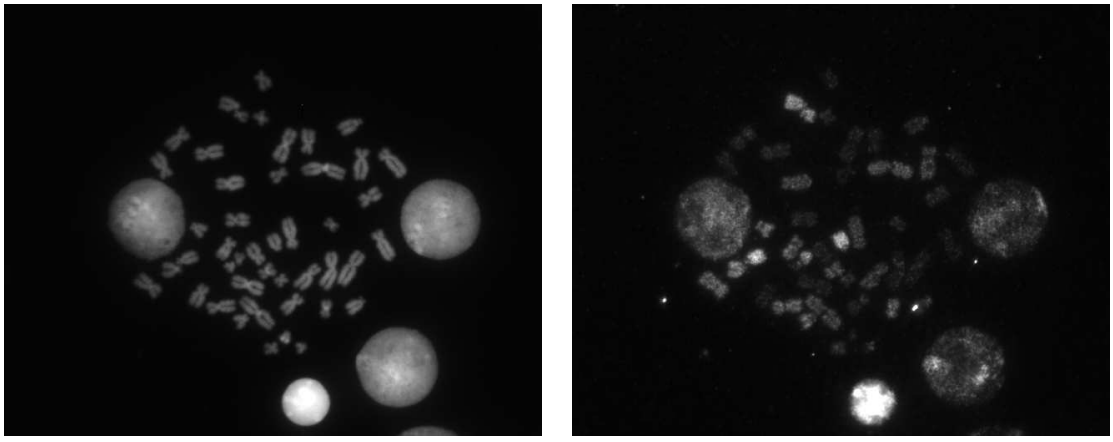
A. M-FISH and cDNA Microarray Imaging Technology

Genome is the smallest element in any living organism that contains all the information of its cellular structures and activities[2]. Living organism of each biological species has its very own genome, and each cell, the basic working unit of the organism, contains a complete copy of the genome. In each cell, the genome is distributed along the **chromosomes**, which are the carriers of entwined DNAs. Segments of the DNA with certain nucleotide sequences are called **genes**, which are expressed or depressed to control the synthesis of protein. The human genome contains about 30,000 genes. M-FISH and microarray imaging technologies are two powerful tools recently developed, which intend to show the properties of genome on the chromosome level and gene level, respectively.

M-FISH imaging is a recently developed technology for molecular cytogenetic analysis [3]. In contrast to the conventional single-staining-based methods, M-FISH specimens are obtained by simultaneous hybridization with a set of chromosome spe-

cific DNA probes, each labeled with a different combination of fluorescent dyes. M-FISH images are acquired through a fluorescence microscope with a turret of multiple optical filters for imaging each of the individual fluorescent dyes separately. These are visible in different optical wavelengths referred to as spectral channels. Thus an M-FISH image set is comprised of a number of images, each aligned to the coordinates of a reference image by performing image registration. Fig. 1 shows two (out of six) channels of a typical M-FISH image set.

M-FISH technology enables multi-color karyotyping thanks to the combination of multiple fluorescent dyes used in the chromosome specific DNA probes. Furthermore, it facilitates unambiguous detection of target-specific chromosomal alterations in human and other mammalian cells, which is especially useful for elucidation of subtle or complex chromosomal rearrangements [4]. For these reasons, M-FISH technology has been increasingly used for the diagnosis of genomic abnormalities in the rapidly growing field of cancer cytogenetics.



(a) (b)

Fig. 1. Two (out of six) channels of a typical M-FISH image set "A0101X" with size $645 \times 517 \times 6$. (a) DAPI channel. (b) Texas Red channel.

cDNA microarray technology is a hybridization-based process that can quan-

titatively characterize the relative abundance of gene transcripts [5, 6]. Contrary to the conventional methods, microarray technology promises to monitor the transcript production of thousands of genes or even the whole genome simultaneously. It thus provides a new and powerful tool for genetic research and drug discovery. To produce cDNA microarrays, the mRNA of the control and test samples are first reverse-transcribed into cDNA, and fluorescently labeled with different dyes (typically red and green). Then the fluorescent targets are mixed and allowed to hybridize with gene-specific cDNA clones printed in an array format on a glass microslide. Finally by scanning the microslide with a laser and capturing the photons emitted from different dyes into different channels with a confocal fluorescence microscope, a two-channel 16-bit microarray image is obtained, in which the pixel intensities reflect the level of mRNA expression. Fig. 2 shows a portion of a typical microarray image in RGB composite format, where the red and green channels correspond to the two channels of the microarray image obtained while the blue channel is set to zero. Each round spot in the figure corresponds to the hybridization site of a certain gene. With the techniques from various areas like image processing, classification, clustering, statistical data analysis, etc., cDNA microarray images can shed light on the complex genetic regulation rules long sought by the biologists and clinicians.

B. Issues of Genomic Image Processing

Genomic image processing can be roughly categorized into three major areas: *processing*, *compression*, and *analysis*. Usually data analysis cannot be performed prior to image processing, while compression sometimes can be performed separately. Basic image processing tasks mainly consist of procedures such as geometric adjustment, noise filtering, segmentation and enhancement. As the very first step of genomic im-

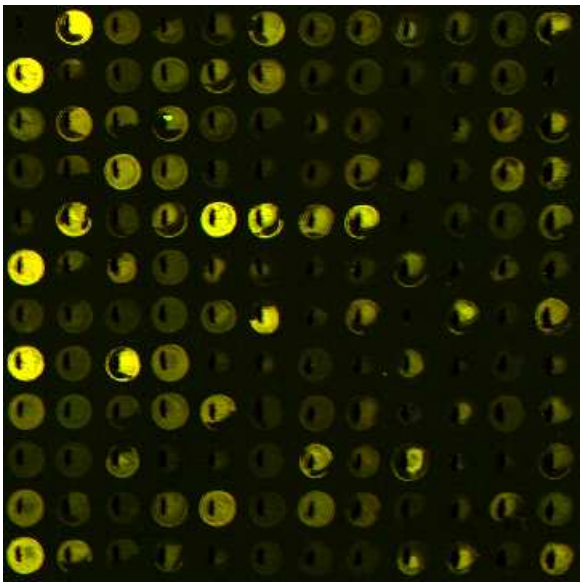


Fig. 2. Part of a typical cDNA microarray image in RGB composite format.

age processing, its accuracy is critical to the reliability of subsequent data analysis. Although many genomic images can be analyzed by the biologists directly after processing, data analysis methods can further help the biologists understand the data from different aspects, and establish a possible link between the data and their biological meanings. For example, classification can associate the data with certain biological functions/diseases, hence help the clinicians make diagnosis decisions. Clustering can be used to obtain a holistic view of the data, and to seed a feature selection algorithm for classification. And genetic regulatory networks can help construct the dynamic system involving different genes and suggest the potential medical interventions/treatments. Besides processing and analysis, image compression is another important issue. The images obtained through expensive biological experiments with precious samples need to be archived for later process and double-check, or even being revisited in the future with much advanced technologies. Moreover, current genomic imaging technologies become more and more involved with parallel techniques which

endeavor to provide as much as possible information to the biologists in one shot. For example, both M-FISH and cDNA microarray apply simultaneous hybridizations with multiple fluorescent dyes across a whole set of chromosomes/genes. These factors will significantly increase both the number and the size of the genomic images to be archived. Thus efficient image compression algorithms are highly desired.

It is impractical to perform a thorough research on all the issues. Hence only selected problems in each category are studied in this dissertation:

- **M-FISH image compression**: Although image compression[1] techniques generally fall into two categories: lossy and lossless compression, lossless compression is preferred in most medical applications due to the possibility of information loss associated with lossy compression[7]. Current method for archiving M-FISH images is to store them channel by channel by losslessly compressing each channel using techniques such as Lempel-Ziv-Welch (LZW) coding [8, 9].

  However, M-FISH image has an important characteristic that is distinct from many other types of medical images: it has a foreground which includes information essential to the diagnosis or analysis, and can be viewed as the region of interest (ROI). On the other hand, the remaining background only provides some reference information. This property opens a door to ROI coding, which encodes the foreground and background independently. In this work we will design a wavelet-based ROI coding scheme for M-FISH image.

- **Microarray image processing**: Unlike most other medical images, the information of microarray images lies in the intensity of each spot, which is not intended to be analyzed under visual inspection. Thus the signals in microarray

[1]We use image coding and image compression interchangeably in this dissertation.

images must be estimated with appropriate image processing procedures before any analysis is taken. In this work we will design an efficient microarray signal estimation scheme which can perform a series of basic image processing functions, including segmentation and background adjustment, to estimate signals for later analysis.

Besides signal estimation, owing to the large data volume associated with microarray images (each typically takes about 15MB to store), highly efficient compression is necessary. Hence in this work, we will design a good progressive compression scheme that provides sufficiently accurate genetic information for data analysis at low bit-rates, while still ensuring good lossless compression performance.

- **Expression-based classification**: Classification via gene expression level estimated from the microarray images requires designing a classifier that takes a vector of gene expression levels as input features, and outputs a class label, which predicts the class containing the input feature vector. Given the joint feature-label distribution, increasing the number of features always results in decreased classification error; however, this is not the case when a classifier is designed via a classification rule from sample data. Typically, for fixed sample size, the error of a designed classifier decreases and then increases as the number of features grows. The problem is especially acute when sample size is very small and the potential number of features is very large, which are exactly the cases encountered in expression-based classification, where the typical sample size is well under 100, and the available gene expression levels usually up to thousands. Thus it is crucial to obtain a general understanding of the range of feature-set sizes that can provide good performance for a particular

classification rule at certain sample size. In this study we will investigate this relationship for various classifiers.

C. Organization of the Dissertation

The major work accomplished in this dissertation consists of three parts: 1) M-FISH image compression; 2) microarray image processing; 3) determination of the optimal feature size as a function of sample size for expression-based classification.

Chapter II discusses the M-FISH image compression where a new coding scheme, the embedded M-FISH image coding (EMIC), is presented. We first review the shape-adaptive integer wavelet transforms and the object-based bit-plane coding which can generate separate embedded bitstreams that allow continuous lossy-to-lossless compression of the foreground and background. Then we propose a method of designing an optimal context model for the bit-plane coding that specifically exploits the statistical characteristics of M-FISH images in the wavelet domain. Experiments have been done to compare our proposed scheme with other popular schemes like LZW coding, JPEG-LS and JPEG-2000.

In Chapter III we target at the microarray image processing for signal estimation and image compression. We present Microarray BASICA: an integrated image processing scheme including tools like segmentation, background adjustment and image compression for cDNA microarray images. For the signal estimation part, we first present a fast Mann-Whitney-test-based segmentation algorithm, followed by the post-processing procedure, and finally the background adjustments. For the image compression part, we first introduce a new distortion measurement for cDNA microarray image compression and then present a coding scheme by modifying the embedded block coding with optimized truncation (EBCOT) algorithm [10].

Chapter IV investigates the relationship between the optimal number of features and sample size for various classifiers in expression-based classification. Both parametric and non-parametric classifiers are discussed. First we provide an analytical approach for the quadratic discriminant analysis (QDA) based on the statistic representation derived by MacFarland and Richards [11]. Then for linear discriminant analysis(LDA) and non-parametric classifiers, we take advantage of the massively parallel computation and perform simulations on the carefully designed distribution models and real patient data.

In Chapter V, we summarize the dissertation on the accomplished works and provide a perspective for the future research in genomic image processing.

D. Main Contributions

- Developed a wavelet-based progressive coding scheme for highly efficient compression of M-FISH images. To achieve this, a new context model design method is proposed.
- Designed a microarray image processing scheme, which performs efficient signal estimation and image compression.
- Found an analytic method to determine the optimal number of features at different sample sizes for QDA classifier.
- Studied the optimal number of features as a function of sample size for various classifiers based on both well-designed distribution models and real patient data. A reference web-site is established to provide a resource for the community in assessing the feature-set sizes.

CHAPTER II

M-FISH IMAGE COMPRESSION*

This chapter presents a new wavelet-based image coder specifically designed for M-FISH image compression. The chapter starts with a brief introduction of the current achievements in wavelet-based image coding, especially in medical image compression. Then our new scheme, embedded M-FISH image coding (EMIC), is discussed in detail.

A. Wavelet-based Medical Image Coding Schemes and M-FISH Image Compression

Image compression techniques generally fall into two categories: lossy and lossless compression. Although lossy compression can achieve higher compression ratios, medical diagnosis is often compromised with its usage due to the information loss [7]. Thus lossless compression is preferred in most medical applications. The current method for archiving M-FISH images is to store them channel by channel by losslessly compressing each channel as tiff image using techniques such as Lempel-Ziv-Welch (LZW) coding [8, 9].

However, LZW coding of M-FISH images fails to exploit either the two-dimensional (2-D) pixel correlation within each channel or the dependencies among different channels (each chromosome is located in the same spatial position across different channels within an M-FISH image set.) This suggests that standard 2-D JPEG [13] or JPEG-2000 [14] coding would outperform LZW coding and that new 3-D wavelet-based coding techniques [15]-[19] could further improve M-FISH image compression.

Shapiro's embedded zerotree wavelet (EZW) coder [20] and the later work by

*©2004 IEEE. Reprinted, with permission, from "Wavelet-based compression of m-fish images", J. Hua, Z. Xiong, Q. Wu, and K. R. Castleman, IEEE Trans. on Biomedical Engineering, to appear.

Said and Pearlman on set partitioning in hierarchical trees (SPIHT) [21, 22] revolutionized the field of wavelet image coding. The new JPEG-2000 standard is based on a scheme called embedded block coding with optimal truncation (EBCOT) [10]. Inspired by the success of wavelet image coding, several authors have extended the existing frameworks to 3-D medical volumetric data compression [16, 17, 18, 19, 23, 24], achieving better results than those from the 2-D approaches [25, 26] and the early work of 3-D wavelet-based medical image compression [15]. Among them, 3-D extensions of SPIHT and EBCOT, namely 3-D SPIHT [27] and 3-D embedded subband coding with optimal truncation (3-D ESCOT) [28] achieve the best coding performance published so far in the literature [17]. An attractive feature of the wavelet-based approach is that, with an integer wavelet transform, one can generate a single *embedded bitstream*[1] that allows progressive lossy-to-lossless compression.

M-FISH images have an important characteristic that is distinct from many other types of medical images: the chromosome regions (the regions of interest to cytogeneticists for evaluation and diagnosis), which are identical among all channels, are well determined and segmented prior to the storage of each image set. These chromosome regions provide diagnostic information and should be losslessly compressed. On the other hand, the remaining background images, which may contain cell nuclei and stain debris, are kept as well in routine cytogenetics lab procedures for specimen reference rather than for diagnostic purposes. Since they usually provide little useful information, lossy compression for them is acceptable. M-FISH images can thus be viewed as consisting of two types of regions of interest (ROI): foreground objects (chromosomes) and background objects (interphase nuclei and stain debris, etc.).

[1] An embedded bitstream has the property that each additional bit improves the quality of the decoded images and that the whole bitstream can be truncated at any point to provide a set of decoded images with quality commensurate with the bit-rate.

Consequently, regions-of-interest coding [29] should be used to treat the foreground and background objects differently (e.g., lossless coding of the foreground objects and lossy-to-lossless coding of the background objects). In [30], an efficient wavelet-based regions-of-interest coding scheme is already proposed for lossy-to-lossless compression of both the foreground and background objects of single-channel chromosome images. Hence it is natural to apply wavelet-based regions-of-interest coding to M-FISH image compression.

B. Embedded M-FISH Image Coding (EMIC)

In this section we introduce wavelet-based embedded M-FISH image coding (EMIC). EMIC seeks to encode M-FISH images adaptively with respect to the image content. Recall that each M-FISH image set can be classified into two types of ROI: foreground objects and background objects. Lossless compression is always needed for the foreground objects as they include all the chromosomes, which are essential to cytogeneticists' evaluation and diagnosis. Lossy compression is acceptable in most cases for the background objects, which contain little diagnostic information. EMIC goes a step further by providing lossy-to-lossless compression for both the foreground and background objects.

Fig. 3 depicts the block diagram of the encoder in EMIC. A set of M-FISH images is first segmented into the foreground and background objects. This is followed by shape coding of the segmentation mask shared by all image channels. Then we apply *critically sampled* integer wavelet transforms to both the foreground and background objects, and encode the shapes of the objects as a small header in the bitstream. After the transforms, object-based bit-plane coding is employed to generate one lossless bitstream for the foreground objects and another layered lossy-to-lossless bitstream

for the background objects. In forming the final encoded bitstream, we follow the syntax that the bitstream generated by shape coding goes first, followed by those corresponding to the foreground objects and the background objects, respectively. The encoding procedure is reversed in the decoder. Although we only aim for lossless compression of the foreground and lossy-to-lossless compression of the background, lossy compression of the foreground objects can also be achieved in EMIC by simply decoding at lower bit-rates. The lossy mode is desirable in applications requiring progressive image transmission, such as telemedicine and fast searching and browsing of M-FISH images. The rest of this section describes different components of EMIC in detail.

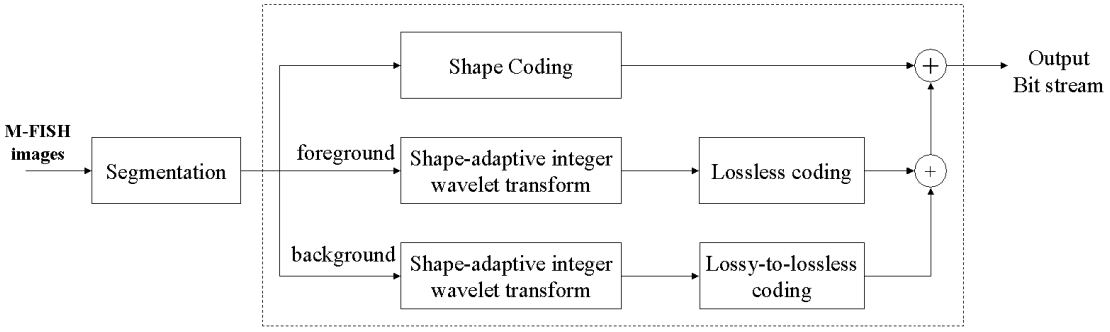


Fig. 3. Block diagram of the encoder in EMIC for M-FISH image compression.

1. Segmentation and Shape Coding

Before object-based bit-plane coding, segmentation must be performed to delineate the foreground objects from the background objects. EMIC can either use an existing segmentation mask generated interactively under the supervision of cytogeneticists, or obtain it through an adaptive thresholding algorithm (e.g., [31]) applied on the DAPI channel of each M-FISH image set.

Different spectral channels of an M-FISH image set share the same segmentation mask. An 8-connected differential chain code [31] is used to compress the segmentation mask. Shape coding of the segmentation mask typically costs about 2.5 kbytes per M-FISH image set. Compared to the average lossless compression results on the foreground objects shown later, this overhead due to shape coding is nominal.

2. Integer Wavelet Transform

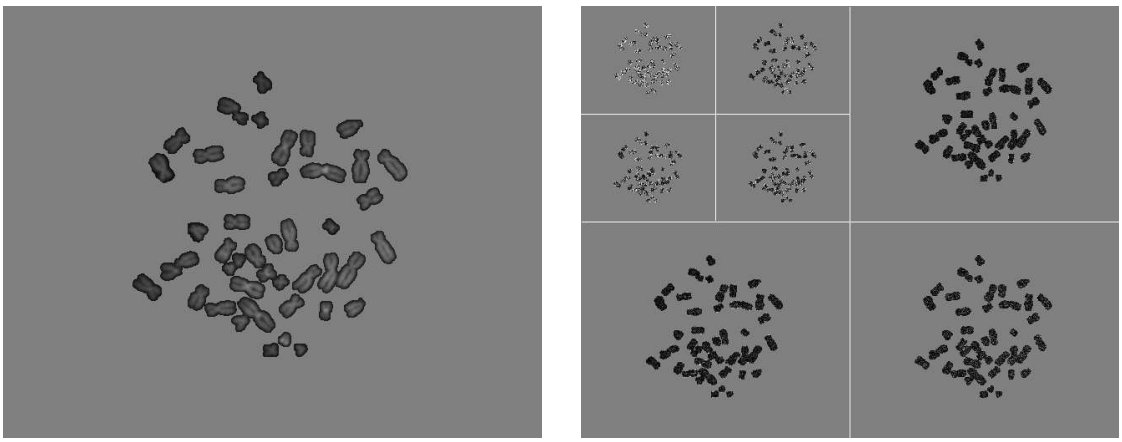
It was shown in [32] that every finite impulse response wavelet or filter bank can be decomposed into lifting steps. In addition to achieving as much as a two-fold speed-up over filtering-based implementations, the lifting-based approach also makes it very easy to have an integer-to-integer mapping, which is a must for lossless image compression [33]. Different wavelet filters are compared for lossy image compression in [34] and lossless image compression in [35]. In general, the 5/3 filters [33] outperform other wavelet filters for lossless compression, while the Daubechies 9/7 filters [36] are the overall best for lossy compression[2]. As reported in [33, 35, 37], different wavelet filters excel at different types of images, hence it was not clear which filters are the best for M-FISH images. Thus in Section C we evaluate the 5/3 and 9/7 filters along with seven other commonly used filters [35], i.e., S+P, (2+2,2), (4,2), (2,4), (6,2), (4,4) and 2/6 filters, to experimentally determine the best among this augmented set of filters for M-FISH image compression.

a. 2-D Shape-adaptive Integer Wavelet Transform

After the segmentation of M-FISH images, both the foreground and background objects are arbitrarily shaped, which demands shape-adaptive integer wavelet trans-

[2]These filters are chosen as the default filters for lossless and lossy image compression, respectively, in JPEG-2000 [14].

forms. In EMIC, we use odd-symmetric extensions over the ROI boundaries [38]. Fig. 4 shows the foreground objects in Fig. 1 (a) and their two-level critically sampled integer wavelet transforms via lifting. It is easy to see that a segmentation mask in the image domain induces a mask for each subband in the transform domain. This wavelet-domain segmentation mask will be used later in the stage of object-based bit-plane coding.



(a) (b)

Fig. 4. Wavelet representation of the foreground objects. (a) The foreground objects of Fig. 1 which include all the chromosomes. (b) The wavelet-domain coefficients after two-level critically sampled integer wavelet transform of the foreground objects.

b. 3-D Integer Wavelet Transform Structure

In 3-D wavelet video coding, we usually use the same wavelet filters in all three dimensions to perform separable wavelet decompositions for both the foreground and background objects. For each object, the 2-D spatial transform and spectral transform (along spectral channels) are done separately by first performing a 2-D dyadic wavelet decomposition on each channel, and then performing a 1-D wavelet decomposition

along the resulting channels[3]. After the transform, there are eight different types of subbands (e.g., LLL, LLH, LHL, HLL, LHH, HLH, HHL, and HHH bands, where the three alphabets from left to right denote the horizontal, vertical, and spectral dimension, respectively.) Similar to JPEG-2000, after transposing some subbands, we can finally end up with only four types (e.g., LLL, LHL, HHL, and HHH bands.)

From our experiments, we find that none of the wavelet filters previously mentioned can efficiently exploit the correlation across different channels. This can be partially explained by the noticeable difference in the average foreground pixel values across these channels (see Fig. 1). This does not mean those cross-channel co-located pixels are not correlated, they actually are as they correspond to the same set of chromosomes. It merely means that 1-D spectral transform across different M-FISH channels is not an efficient way of exploiting this correlation. Hence in EMIC, we allow the option of not performing the 1-D spectral transform after the 2-D spatial transform of each channel. This option eliminates the spectral highpass bands, such as the HHH bands, and relies on efficient context modeling to exploit the correlation among all six channels in adaptive arithmetic coding [39].

3. Fractional Bit-plane Coding

After the shape-adaptive integer wavelet transform, the wavelet coefficients are compressed with bit-plane coding. EMIC employs the bit-plane coding scheme used in embedded wavelet video (EWV) coding [40], which was originally designed for low bit-rate video coding. Below we briefly review the fractional bit-plane coding scheme in EWV which is a 3-D extension of 2-D EBCOT [10] in the JPEG-2000 standard. It offers high compression efficiency and other functionalities (e.g., error resilience and

[3]The pixel mean of ROIs is subtracted off before wavelet decomposition, as is done in coders like SPIHT [21] and 3-D SPIHT [27].

random access) for image coding. Major components of this scheme are discussed below.

Coding primitives: The state of a coefficient is initially set to insignificant and changed to significant when the coefficient's first non-zero bit-plane value is encoded. Depending on the states of the nearby coefficients, the current coefficient's binary information bit at each bit plane is coded using one of the following three primitives: a) Zero coding (ZC): When a coefficient is not yet significant in the previous bit planes, this primitive is used to code whether it becomes significant or not in the current bit plane. b) Sign coding (SC): Once a coefficient becomes significant in the current bit plane, SC is called to code its sign. c) Magnitude refinement (MR): This primitive is used to code the bits of a coefficient if it is already significant.

Fractional bit-plane coding: Using the above three coding primitives in bit-plane coding, one can generate an embedded bitstream for each subband. Specifically, the coding procedure consists of the following three consecutive passes in each bit plane: a) Significance propagation pass: This pass processes coefficients that are not yet significant but have a preferred neighborhood. We use the ZC and SC primitives to code these coefficients' significance information and, if necessary, their sign bits. b) Magnitude refinement pass: Coefficients that became significant in previous bit planes are coded in this pass. The binary bits are coded by the MR primitive. c) Normalization pass: Processed in this pass are coefficients that are not coded in the previous two passes; these coefficients are not yet significant, so the ZC and SC primitives are applied. Each of the above passes processes one fractional bit plane in the natural raster-scan order.

Bitstream construction and scalability: In this stage, bitstreams corresponding to different subbands will be truncated and multiplexed into a final bitstream. First, an operational R-D curve for each subband can be obtained through the fractional bit-

plane coding. Then, given a target bit-rate R_0, optimal rate allocation, i.e., minimum distortion, over all subbands is achieved when operation points on all operational R-D curves have equal slope λ. Lossless coding is achieved by encoding all bit planes, i.e., setting λ to zero. The bitstream with multiple layers is obtained by breaking each subband's bitstream into multiple layers for different rates, and multiplexing them. Since each subband is coded separately, it can achieve scalability in both rate and resolution with great flexibility.

a. Object-based Coding

The extension of fractional bit-plane coding to shaped-adaptive coding is straightforward and efficient. The wavelet-domain representation of a typical M-FISH image set's foreground objects is shown at Fig. 4 (b). Because the shape-adaptive integer wavelet transform is critically sampled, the number of wavelet coefficients is the same as that in the original foreground objects. Using the wavelet-domain segmentation mask, we can easily decide whether a coefficient belongs to the object. If any neighbor of that coefficient falls outside the object, we just set that neighboring coefficient's value to zero and never code it. The object-based EMIC scheme is inherently better than 3-D SPIHT [27], whose rigid cubic zerotree structure is almost surely inefficient in covering an arbitrarily shaped object.

4. Wavelet Coefficient Context Modeling

The generic context model in [28, 40] for arithmetic coding is designed for natural video sequences. However, for any special class of images like M-FISH data, a generic context model cannot fully exploit the peculiarities that are data specific. Thus, designing a particular context model for M-FISH images is essential for better arithmetic coding performance. In this section we focus on optimizing the context model

in EMIC. We first describe a general approach to optimal context modeling for a given data source. We then explain how to apply this approach to the context model design problem in EMIC.

a. A General Approach of Optimal Context Modeling

Consider a data sequence $x_1, x_2, \ldots, x_N$ drawn from alphabet set $\mathcal{X}$ of a stationary random process X. For each sample x_i, one can form its context model C using its preceded samples, i.e., $x_{i-1}, ..., x_1, x_0$. Assuming X is an m-th order Markov process, which is reasonable for wavelet-domain image coefficients[4], its context model C for x_i can be naturally made up of the m symbols $x_{i-1}, ..., x_{i-m}$. Then for large N, the minimum code length (in bits per symbol) is the m-th order conditional entropy, $H(X|C)$, of X given C [41].

Rissanen has shown in [42] that for a given K-parameter context model C, the minimum model adaptation cost is $\Delta_C = \frac{1}{2}(K/N)\log_2 N$ per symbol. Although a context model with higher K decreases $H(X|C)$ [41], it also induces a higher context model adaptation cost Δ_C. For a binary Markov process generated from the raster-scan of bit planes, K is equal to the number of contexts, i.e., 2^m. One way to limit the model cost is to quantize the 2^m contexts into k, $k \ll 2^m$, with $Q(C) \in \{\bar{c}_1, \bar{c}_2, \ldots, \bar{c}_k\}$. Thus the aim of optimal context modeling is to minimize the average codelength

$$L_c(X) = H(X|Q(C)) + \Delta_{Q(C)}. \tag{2.1}$$

To find the optimal context model, one must determine the optimal number of contexts k, the context decision region $A_i = \{C : Q(C) = \bar{c}_i\}$ for each context $\bar{c}_i$, and the corresponding conditional probability $p(X|Q(C) = \bar{c}_i)$, for $1 \le i \le k$. A direct

[4]Although wavelet coefficients are almost uncorrelated, there still remains high-order dependencies among them.

approach is to begin with a small k (e.g., $k = 2$); for each k, find the optimal context model and compute the corresponding $L_c(X)$; increase k until the model adaptation cost $\Delta_{Q(C)}$ becomes dominant, i.e., until $L_c(X)$ stops decreasing or even increases for several successive k's.

For a given k, the key to optimal context modeling is to find the optimal quantizer $Q(C)$ that minimizes $H(X|Q(C))$ in Eq. (2.1). Since $H(X|Q(C)) \geq H(X|C)$ due to the convexity of the entropy function $H(\cdot)$, it has been shown in [43] that the optimization procedure is equivalent to minimizing

$$\begin{aligned} H(X|Q(C)) - H(X|C) &= \sum_c p(c) D(p(X|c) \| p(X|Q(c))) \\ &= D(p(X|C) \| p(X|Q(C))), \end{aligned} \tag{2.2}$$

where $D(p(X|c)\|p(X|Q(c)))$ is the relative entropy between $p(X|c)$ and $p(X|Q(c))$ under a given context c, and $D(p(X|C)\|p(X|Q(C)))$ is the conditional relative entropy (Kulback-Leibler distance [41]) between the conditional distribution of X given C and the conditional distribution of X given $Q(C)$.

There is no close-form solution to the problem in Eq. (2.2). However, if $D(p(X|C)\|$ $p(X|Q(C)))$ is viewed as the cost function and $D(p(X|c)\|p(X|Q(c)))$ as the distance measure, then it is similar to a hard clustering problem [44]: cluster the contexts into k distinct decision regions to minimize the cost function. Thus the K-means algorithm in classification (or the LBG algorithm in vector quantization [45]) which iteratively updates the decision regions and the conditional probability distributions can be used to find a local-optimal solution. It is shown in [43] that for any context c with the optimal cluster $Q(c) = \bar{c}_i$, updating its decision region has to follow the condition

$$D(p(X|c)\|p(X|Q(c))) \leq D(p(X|c)\|p(X|Q'(c))) \tag{2.3}$$

for any $Q'(c) \neq Q(c)$. Here we point out that updating the conditional probability distribution $p(X|Q(C))$ is based on another condition that, for the optimal $p(X|Q(C) = \bar{c}_i)$ of decision region A_i,

$$\sum_{c \in A_i} p(c) D(p(X|c) \| p(X|Q(c))) \leq \sum_{c \in A_i} p(c) D(p(X|c) \| p'(X|Q(c))) \tag{2.4}$$

for any $p'(X|Q(c)) \neq p(X|Q(c))$. This condition can be easily proved from Eq. (2.2). Below we give a detailed description of the general context clustering algorithm.

Cluster 2^m contexts into k contexts

- Initialization: Choose an initial set of conditional probability distributions $p(X|Q(C) = \bar{c}_1), \ldots, p(X|Q(C) = \bar{c}_k)$.
- Repetition:
  - Update the decision regions: for each context c, let

$$Q(c) = \arg\min_{\bar{c}} D(p(X|c) \| p(X|Q(c) = \bar{c})). \tag{2.5}$$

  - Update the conditional probability distributions: for each decision region A_i, let

$$\begin{aligned} p(X|Q(C) = \bar{c}_i) &= \arg\min_{q(X|\bar{c}_i)} \sum_{c \in A_i} p(c) D(p(X|c) \| q(X|\bar{c}_i)) \\ &= \sum_{c \in A_i} p(c) p(X|c) / \sum_{c \in A_i} p(c), \end{aligned} \tag{2.6}$$

    where the second equation follows the results obtained through the Lagrange multiplier method under the constraint $\sum_x q(x|\bar{c}_i) = 1$. Note that the optimal probability distribution is the centroid of the current region, which confirms the suggestion in [43].
  - Evaluation: compute the cost function $D(p(X|C) \| p(X|Q(C)))$ under the current context model parameters.

- Stopping criterion: continue until the cost does not change between two successive iterations.

With this approach, the 2^m contexts are clustered into k contexts to form the optimal context model.

b. Optimal Context Modeling for EMIC

The general approach to context modeling described above assumes the input data sequence is m-th order Markov. In practice, m is not known *a priori*. Only by determining m first can we correctly form the context model for the current wavelet coefficient.

To achieve this, we consider 18 samples in the 3-D neighborhood of the current coefficient (see Fig. 5). We first put these 18 neighbors into 6 categories: immediate horizontal neighbors (h), immediate vertical neighbors (v), immediate spectral neighbors (s), horizontal-vertical diagonal neighbors (d_{hv}), horizontal-spectral diagonal neighbors (d_{hs}) and vertical-spectral diagonal neighbors (d_{vs}).

We compute the correlation between the current coefficient and those in each category in different wavelet subbands. The correlation coefficients obtained over eight randomly picked training image sets are shown in Table I. We single out the DAPI channel from other channels because the correlation pattern of DAPI channel is significantly different from other channels. One interesting observation from Table I is that the correlation between the current coefficient and the immediate spectral neighbors is around 0.3 for all subbands and channels. Although these numbers are probably over-estimated due to some isolated pixels with large values, they indicate that there are positive correlations among channels that can be exploited. Another observation is that for all channels except DAPI, the intra-channel correlation is much

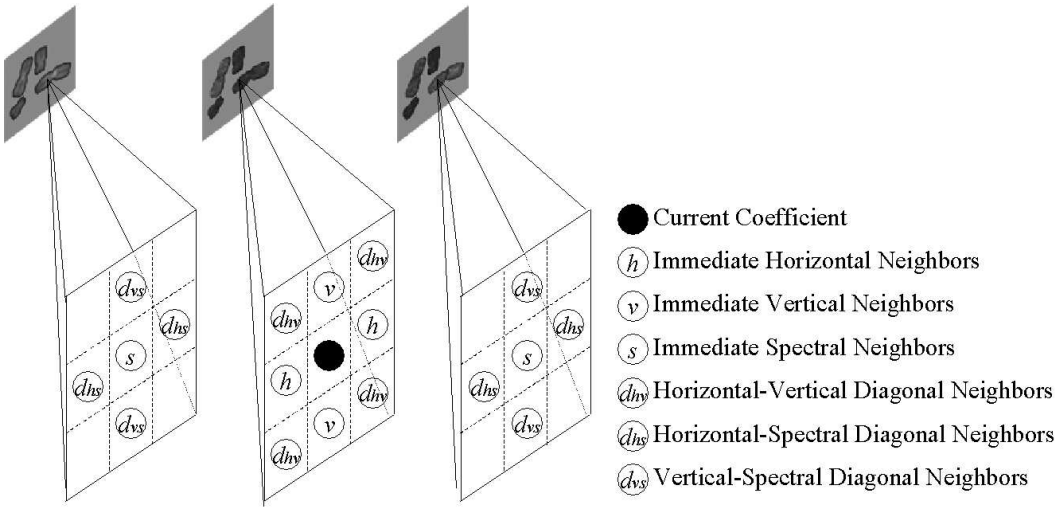


Fig. 5. The 18 8-connected neighbors are categorized into 6 types of neighbors. These neighboring coefficients and the current coefficient are spanned in three consecutive channels, e.g. DAPI, Spectrum Green (where the current coefficient locates), and Spectrum Orange.

higher in the LLL band and drastically lower in other subbands. We also see that the two categories of neighbors d_{hs} and d_{vs}, i.e., horizontal-spectral diagonal neighbors and vertical-spectral diagonal neighbors, are almost uncorrelated with the current coefficient in all subbands. Thus we drop them and form the context model with the 10 coefficients in the remaining four categories for the ZC primitive. For coding the current coefficient, however, the context model includes all coded bits of the 10 neighboring coefficients. For the SC primitive, like many other coders [14, 46], only the six direct neighbors, i.e., h, v and s neighbors, are involved.

The iterative scheme described in the general approach requires the information of conditional probability distribution $p(X|c)$ for each context c. However, in practical implementation, this distribution can only be obtained from limited training data. The situation becomes even worse for wavelet-based M-FISH image coding, where the foreground objects are further decomposed into different subbands. The $p(X|c)$

Table I. Correlation coefficients between the current coefficient and its neighbors. The (6,2) wavelet filters with three-level decomposition are used. The correlations are averaged over eight randomly selected training image sets. The results under **DAPI** column are obtained when current coefficient is in DAPI channel, and **Others** column when it is in other channels.

	LLL band		LHL bands		HHL bands	
	DAPI	Others	DAPI	Others	DAPI	Others
h	0.268	0.681	0.202	0.004	-0.069	0.014
v	0.250	0.706	-0.311	-0.104	0.063	0.045
s	0.243	0.269	0.358	0.345	0.263	0.287
d_{hv}	0.339	0.595	-0.0733	-0.074	0.121	0.054
d_{hs}	0.017	0.107	0.073	0.028	-0.016	-0.007
d_{vs}	0.056	0.130	-0.080	-0.043	0.031	0.017

estimated under limited training symbols can lead to a context model that has good performances only on training images, a problem similar to *overfitting* in pattern recognition [44]. To avoid this, the total number of contexts should be judiciously chosen to ensure sufficient training symbols in each context. On the other side, since context clustering is an irreversible procedure, special attention must be paid to avoid merging contexts that might belong to different decision regions.

Binary quantization is one way of reducing the context size. A binary-valued state variable $\sigma[i,j,k]$ that characterizes the significance of coefficient $x[i,j,k]$ at position $[i,j,k]$ is introduced. It is initialized to 0 and toggled to 1 when $x[i,j,k]$'s first non-zero bit-plane value is encoded. This value is already used in Sec. B.3 of this chapter to decide which coding primitive to use. It quantizes the coded bits of each coefficient in the context model into a binary value, and hence efficiently reduce the

context size. However, with 10 binary symbols in the context, a total of 2^{10} contexts are still too many to obtain reliable $p(X|c)$ in M-FISH context modeling.

There is no reason to treat coefficients in the same category defined in Fig. 5 differently, we thus let h, v, s, and d_{hv} denote the number of coefficients that are already significant in their own categories. Since the case that most coefficients in one category are simultaneously significant is very rare, we further cap h, v, and s at one, and d_{hv} at two. With this procedure, the context size is reduced to 24 for the ZC primitives. And for the LLL and HHL bands, h and v are merged into $h + v$ and the context size is further reduced to 18. As for the SC primitives, we use the 13 contexts provided in EWV [40]. Although the context size seems relatively small, our experiments show that this is sufficient for coding of the foreground objects.

We point out that the fractional bit-plane coding scheme is actually another type of context clustering. It effectively clusters the contexts into three context state sets, i.e., ZC, SC, and MR primitives, and provides great flexibility by introducing fractional bit planes. It was shown in [47] that the associated performance loss is nominal. Thus our context modeling procedure starts with fractional bit-plane coding and ends with separate optimization of sub-context models for the ZC, SC, and MR primitives. However, whenever there is no confusion, we will still call them context modeling in the sequel.

The general context modeling approach is based on the assumption that the input data sequence is stationary. However, from Table I we see that the data sequence is not stationary (e.g., the DAPI channel and other channels are statistically different). Also the binary sequence generated under the fractional bit-plane coding scheme of EMIC is not stationary among different fractional bit planes.

To ensure the stationarity of the data sequence, we only consider the data from one type of subbands at a time, and treat DAPI channel and other channels in-

dependently. This means that we need to design separate optimal context models for sub-sequences of the data from different subbands and channels. Then for each sub-sequence we convert the binary sequence into a non-binary sequence, where each symbol is made up of the bits from all fractional bit planes, i.e., $x = \{x^1, \ldots, x^B\}$, where B is the number of fractional bit planes, and x^i is the symbol in fractional bit plane i. Note that since each bit is coded only once in one of the three fractional bit planes that form that bit plane, x^i can actually have three possible values: 1, 0 and VOID, where the VOID corresponds to the case when no bit is coded in the current fractional bit plane. It is reasonable now to assume that each new input sequence is stationary. Then by assuming that the probability distribution of x^i depends only on its context, the average codelength can be written as

$$\begin{aligned} E_X[L(X)] &= -\sum_{i=1}^{N}\sum_{X} P(x)\log_2 p(x_i|C) \\ &= -\sum_{i=1}^{N}\sum_{X}\sum_{j=1}^{B} P(x)\log_2 p(x_i^j|C^j) \\ &= \sum_{j=1}^{B} N \cdot H(X^j|C^j), \end{aligned} \tag{2.7}$$

where N now denotes the number of symbols coded in each fractional bit plane. Then the cost function in Eq. (2.2) becomes

$$\begin{aligned} H(X|Q(C)) - H(x|C) &= \sum_{j=1}^{B}\sum_{c} p^j(c) D(p^j(X^j|c)\|p^j(X^j|Q^j(c))) \\ &= \sum_{j=1}^{B} D(p^j(X^j|C^j)\|p^j(X^j|Q^j(C^j))). \end{aligned} \tag{2.8}$$

Although this approach can find the optimal context model for each fractional bit plane, it also scales the total number of contexts by the number of fractional bit planes, which will induce high context adaptation cost. One thus must consider

different fractional bit planes jointly in order to achieve a good balance between the conditional entropy $H(X|Q(C))$ and the context adaptation cost $\Delta_{Q(C)}$.

By observing the training sets of M-FISH images we notice that for each context c, the conditional probability distribution $P(X|c)$ changes smoothly between adjacent fractional bit planes. This implies that different fractional bit planes can use the same optimal context model to reduce context adaptation cost. Thus in optimizing the context model, we let $Q^i(c) = Q^j(c)$, $1 \leq i, j \leq B$.

The iterative optimization procedure in Sec. B.4.a of this chapter is used to design the context model for the ZC and SC primitives in different subbands. For the MR primitive, since the refinement bits are known to be almost uniform, EMIC does not perform any context model optimization and keeps the one used in EWV [40]. Separate context models are designed for the DAPI channel and other channels in the LLL, LHL, and HHL subbands. Thus six context models are obtained for each ZC or SC primitive.

The memory usage of the 3-D context model in EMIC is larger than EBCOT's 2-D model but much smaller than EWV's 3-D model. Compared to EWV's generic context model, EMIC only uses a total of 10 neighbors and 6 tables for the ZC primitive. Thus the look-up tables for the ZC context assignment have a maximum of 6×2^{10} items, which are 128 times smaller than EWV's 3×2^{18} items for three tables, and are about 6 times larger than EBCOT's 2×2^9 items for two tables. Compared to EBCOT and EWV, the look-up tables for the SC primitive in EMIC cost more memory. But since the SC primitive's context tables are much smaller than the ZC primitive's, the overall memory cost is actually determined by the latter. As for the complexity issue, setting up the look-up tables takes little time and once it is done, these tables are easy to use. Thus our M-FISH specific context model design does not bring more complexity to the coding procedure than the general context models

in EBCOT or EWV.

C. Experimental Results

Experiments have been conducted to test the performance of EMIC on a total of 88 different M-FISH image sets from a publicly available M-FISH image database (http://www.adires.com /05/Project/MFISH_DB/MFISH_DB.shtml). Each set has six channels, all with size 645 × 517 and eight bit resolution. These M-FISH image sets belong to the ASI group of test images in the database.

1. Lossless Coding Performance for the Foreground Objects

a. EMIC Results with Different Wavelet Filters and Decomposition Levels

Table II lists the lossless compression results of EMIC with different wavelet filters and decomposition levels. For these tests we randomly selected 8 out of the 88 image sets. We tested all the nine integer wavelet filters mentioned in Sec. B.2.a of this chapter with the level of wavelet decomposition ranging from one to five. From these results we notice that with the same decomposition level, the (2+2,2), (4,2), (4,4), and (6,2) wavelet filters perform closely and they achieve slightly higher compression ratios than the others. For all nine wavelet filters, the compression ratio reaches a peak at the decomposition level of two or three. After that, the compression ratio peaks out and even starts to decrease slightly. Unlike zerotree-based coding schemes, EMIC does not always perform better when the decomposition level increases. This is because the increase in decomposition level results in more small-size subbands. As each subband has its own model-adaptation cost in arithmetic coding, the loss in adaptation cost cancels out the gain from using more decomposition levels at some point. Among the nine wavelet filters and five different decomposition levels, the

(6,2) wavelet filters with a two-level or three-level decomposition performed the best during this test on the eight M-FISH image sets.

Table II. Lossless compression results for the foreground objects of M-FISH images using EMIC with different integer wavelet filters and decomposition levels. The shown compression ratios are in bits /pixel/channel and are averaged over the eight test image sets.

	1 level	2 levels	3 levels	4 levels	5 levels
9/7-F	0.3720	0.3659	0.3655	0.3656	0.3657
(2+2,2)	0.3642	0.3594	0.3596	0.3598	0.3599
(2,2)	0.3693	0.3648	0.3649	0.3651	0.3652
(S+P)	0.3754	0.3681	0.3672	0.3673	0.3674
(4,2)	0.3645	0.3594	0.3594	0.3596	0.3597
(2,4)	0.3708	0.3669	0.3670	0.3672	0.3673
(4,4)	0.3648	0.3599	0.3599	0.3600	0.3602
(6,2)	0.3645	0.3593	0.3593	0.3595	0.3596
2/6	0.3759	0.3691	0.3685	0.3686	0.3687

b. Comparison with Other Lossless Coding Techniques

We have compared EMIC against several popular lossless coding schemes: LZW in WinZip 8.0, JPEG-LS, and JPEG-2000. We used the JPEG-LS Reference Encoder V.1.00 implementation by Hewlett-Packard for JPEG-LS coding and Taubman's Kakadu V2.2 implementation for JPEG-2000 coding. Because the 2-D based JPEG-LS and JPEG-2000 coders cannot compress the multi-channel M-FISH image set as a whole, the six channels in each set were coded separately when these two

coders were used, and the sums of six compressed file sizes are reported.

Since LZW and JPEG-LS can only handle lossless compression of regularly shaped images, we set the background pixels in test images to zero for these coders. For JPEG-2000, because coding of the foreground and the background is not done separately, no lossless reconstruction of the foreground objects can be guaranteed until the whole image set is recovered. Therefore, the same test images with zero background for LZW and JPEG-LS were used for the JPEG-2000 tests to ensure lossless recovery of the foreground objects. Lossless compression results from different coders are summarized in Table III. EMIC turns out to perform much better than the other popular coders under study. It achieves an average saving of 78%, 72%, and 17% over LZW, JPEG-2000, and JPEG-LS, respectively. Note that the result of EMIC already includes the overhead of shape coding of the segmentation mask, which, as described in Sec. B.1 of this chapter, is around 2.5 kbytes, or 0.01 bits/pixel/channel for each M-FISH image set. LZW-based WinZip 8.0 gives the poorest result, mainly because it does not take advantage of the 2-D or 3-D structure of the image data. The performance of JPEG-2000 is not very good either because its wavelet transform is not critically sampled, thus more samples need to be coded in the wavelet domain. JPEG-LS performs better than LZW and JPEG-2000 but is still behind EMIC. Furthermore, in contrast to LZW and JPEG-LS, EMIC is capable of providing a scalable lossy-to-lossless bitstream for a given image set. This progressive coding property is achieved in EMIC's encoding process by inserting truncation points to form a layered bitstream. The decoder can simply stop at any truncation point. Bitstream scalability is desirable in applications requiring progressive image transmission, such as in telemedicine and fast browsing of M-FISH images.

Besides the above popular coding schemes, we also provide results based on EMIC with the generic context model from EWV, which we denote as EWV in Table

III. With the generic context model, EWV is slightly worse than EMIC. These results indicate that the generic context model is capable of providing rather good performance. However, our context model specifically designed for M-FISH images gives even better compression performance.

Table III. Lossless compression results of the foreground objects. The bit-rates shown are in bits/pixel/channel and are averaged over 88 M-FISH image sets. The (6,2) wavelet filters are used with three levels of decomposition in EMIC and EWV.

	LZW	JPEG-2000	JPEG-LS	EWV	EMIC
Bit-rate	0.6030	0.5830	0.3978	0.3411	0.3396

2. Lossy-to-lossless Coding Performance for the Background Objects

a. EMIC Results with Different Wavelet Filters and Decomposition Levels

EMIC allows different choices of wavelet filters and decomposition levels for the foreground and background objects. This is because they are separately coded. Although lossless coding might be needed for the background objects, lossy coding is usually acceptable. The setting for lossless coding of the foreground objects (i.e., the (6,2) wavelet filters with three-level decomposition) might not be the best choice for the background objects. Therefore a comparison of the EMIC performance with different wavelet filters for lossy-to-lossless compression was performed on the same eight sets of M-FISH images used in Sec. C.1 of this chapter.

Comparison results for the nine integer wavelet filters used in this study are shown in Fig. 6 (a). Note that the (2,4) wavelet filters achieve the best PSNRs for lossy coding despite the fact that they are not outstanding for lossless coding. Among

all the nine filter pairs, the (4,4) and (2+2,2) wavelet filters yield relatively good performance for both lossless and lossy coding. Besides the performance differences caused by the choices of wavelet filters, the decomposition level also appears to affect the lossy-to-lossless coding performance. Performance comparisons in PSNR using the (2,4) wavelet filters with different levels of wavelet decomposition and bit-rates are shown in Fig. 6 (b). Unlike the lossless coding experiments in which the best performance is achieved with a two or three-level decomposition, when the bit-rate is relatively low the PSNRs of reconstructed images in these cases keep increasing as the wavelet decomposition level goes up. This is because more energy is compacted into the lowpass subbands as the decomposition level increases, and so M-FISH image sets can be reconstructed with higher PSNRs. When the bit-rate is relatively high, the performance difference due to different decomposition levels diminishes. The incremental gain also becomes smaller from one level to another.

Table IV shows the PSNRs of each channel of a reconstructed M-FISH image set "A0101XY" by EMIC at different bit-rates. Here the (2,4) wavelet filters with a five-level decomposition is used. The background of this M-FISH image set is first coded losslessly at 2.53 bits/pixel/channel. This lossless bitstream is truncated at 0.01, 0.025, 0.05, 0.10, and 0.15 bits/pixel/channel, respectively to obtain decoded images with the PSNRs shown in the table. Note that these bit-rates are for the background objects only. The segmentation mask and the foreground objects of M-FISH images are already losslessly coded with the bit-rate 0.34 bits/pixel/channel. Thus the total bit-rate for the whole image set is the sum of the lossy bit-rate and 0.34 bits/pixel/channel.

Fig. b shows two channels (DAPI and Texas Red) of the original and reconstructed image set "A0101XY". The bit-rates shown are for the background objects only. When reconstructed at 0.01 bits/pixel/channel, the blur of nuclei in the

Table IV. PSNR (in dB) of each channel of M-FISH image set "A0101XY" reconstructed at different bit-rates in bits/pixel/channel. The (2,4) wavelet filters are used with a five-level decomposition.

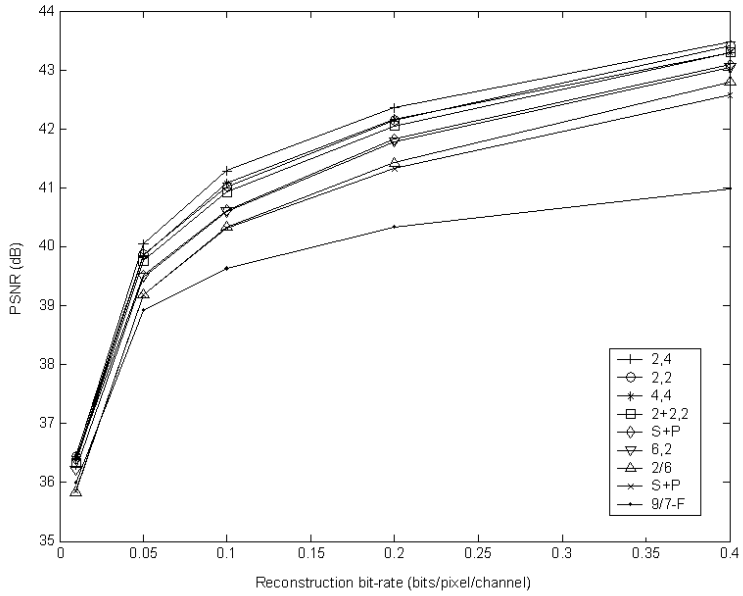
Bit-rate	0.01	0.025	0.05	0.1	0.15
DAPI	35.69	38.00	41.98	43.76	44.64
Green	32.69	36.14	37.71	39.26	40.18
Orange	30.20	34.77	36.33	38.04	39.04
Texas Red	33.25	36.57	38.78	40.89	41.87
Cy 5	34.20	36.79	38.20	39.81	41.02
Cy 5.5	32.54	36.40	38.21	40.49	42.44
Average	33.43	36.45	38.53	40.37	41.53

background objects is noticeable; at 0.025 bits/pixel/channel, the results look much better; at 0.05 bits/pixel/channel, the image quality is reasonably good; at 0.10 bits/pixel/channel, most details in the original images are present in the reconstructed ones; and at 0.15 bits/pixel/channel, the original images and the reconstructed ones become indistinguishable.

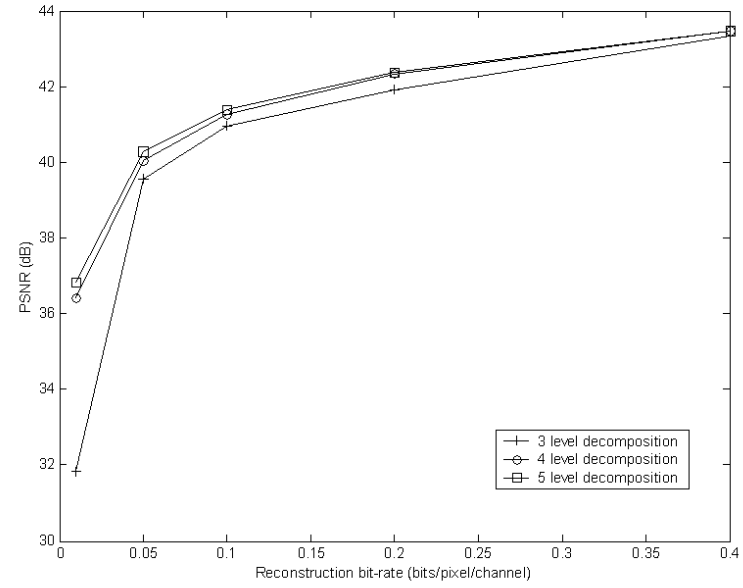
Fig. 7 depicts the bit-rate vs. averaged PSNR over all 88 M-FISH image sets. Again the (2,4) wavelet filters are used with a five-level decomposition and the bit-rate is for the background objects only (the average bit-rate for the foreground is 0.34 bits/pixel/channel, as seen from Table III, which already includes the bit-rate for the segmentation mask). Note that, in order to achieve lossless compression of the background objects, the average required bit-rate is 2.49 bits/pixel/channel. This is much higher than the 0.5 bits/pixel/channel needed on average to produce perceptually good image quality with PSNR at 44.5 dB.

b. Comparison with JPEG-2000

Finally we compare EMIC with JPEG-2000. Recall that JPEG-2000 only supports lossless coding of regularly shaped images. We set the background of M-FISH images to zero (or a constant value in general) for JPEG-2000 to achieve lossless compression of the foreground. By doing so both the foreground and the zero background are perfectly recovered. Table III shows that JPEG-2000 spends 0.58 bits/pixel/channel on average for this purpose, indicating it is not efficient for lossless regions-of-interest coding because bits are wasted coding the zero background. In contrast, EMIC only uses 0.34 bits/pixel/channel on average to achieve lossless compression of the foreground ROI. Given the average bit-rate of 0.58 bits/pixel/channel required by JPEG-2000 to achieve lossless foreground coding, EMIC can not only code the foreground lossless with 0.34 bits/pixel/channel but also produce on average PSNR around 42.5 dB for lossy coding of the background using the remaining 0.24 bits/pixel/channel. Because EMIC achieves bit-rate savings from lossless compression of the foreground, it can afford lossy compression of the background, as opposed to having to flatten the background in order to ensure lossless foreground compression with JPEG-2000. Therefore EMIC clearly provides superior coding capabilities to what JPEG-2000 does for M-FISH image compression.



(a)



(b)

Fig. 6. PSNR performance of EMIC under different wavelet filters and decomposition levels. The results shown are the average PSNRs of eight sample M-FISH image sets reconstructed at different bit-rates. (a) Comparison between the nine wavelet filters, all with four-level decomposition. (b) Comparison between different levels of decomposition using the (2,4) wavelet filters.

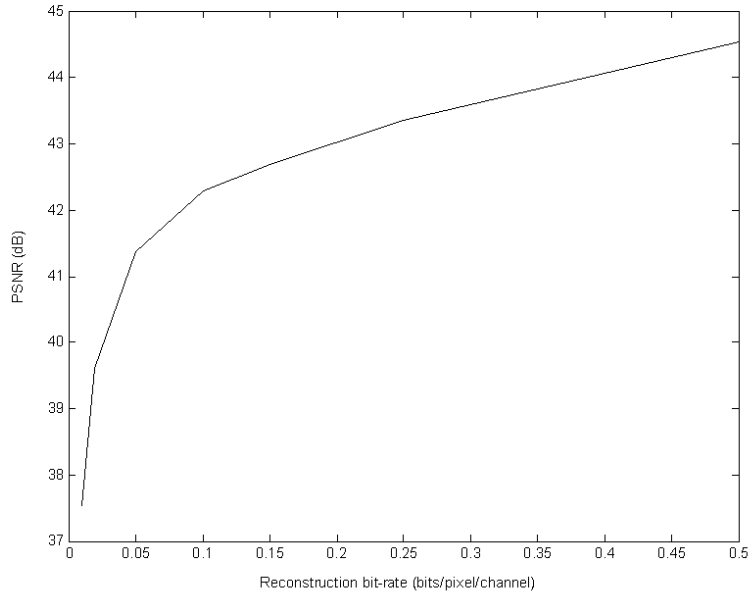
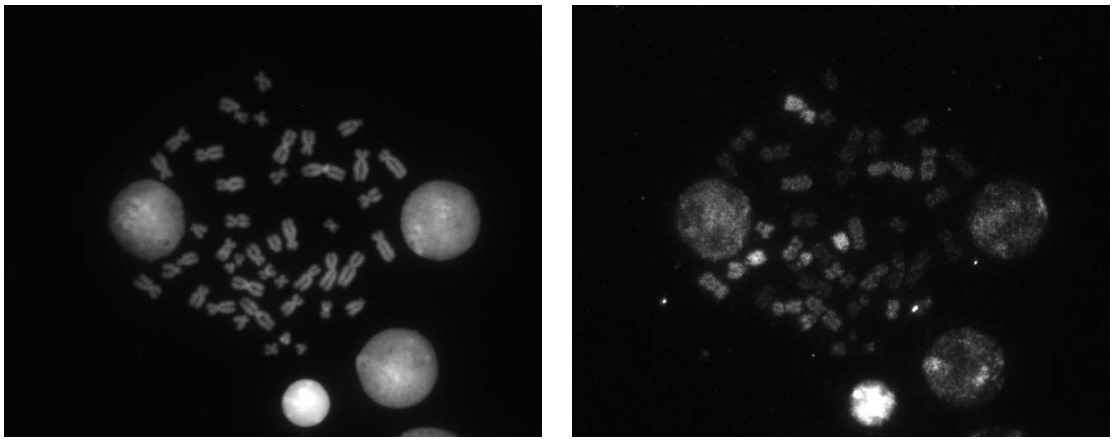
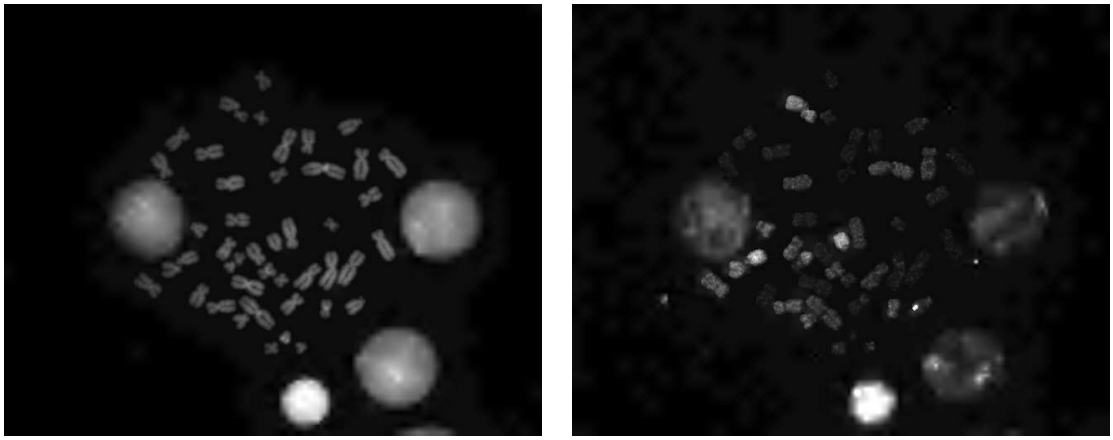


Fig. 7. Average PSNRs from using EMIC for lossy coding of the background objects at different bit-rate. The results are averaged over 88 M-FISH image sets and computed on the background objects only.

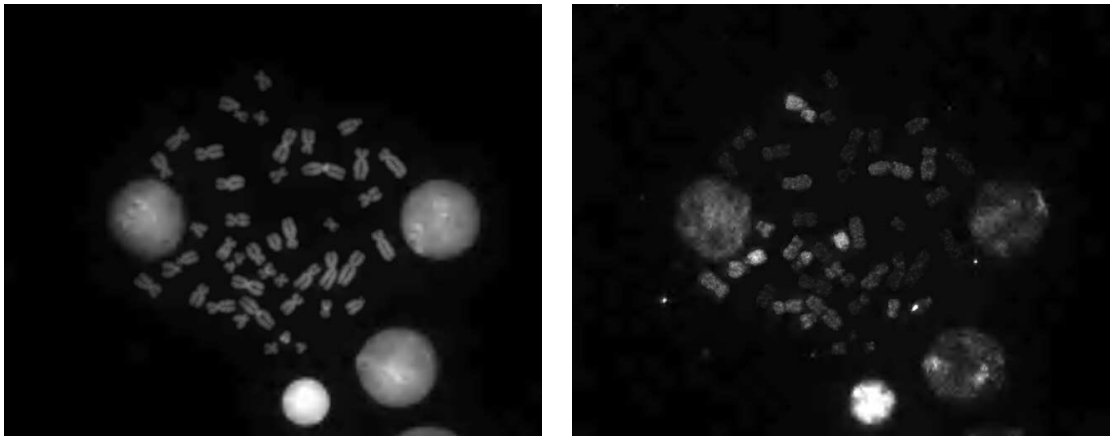


(a)

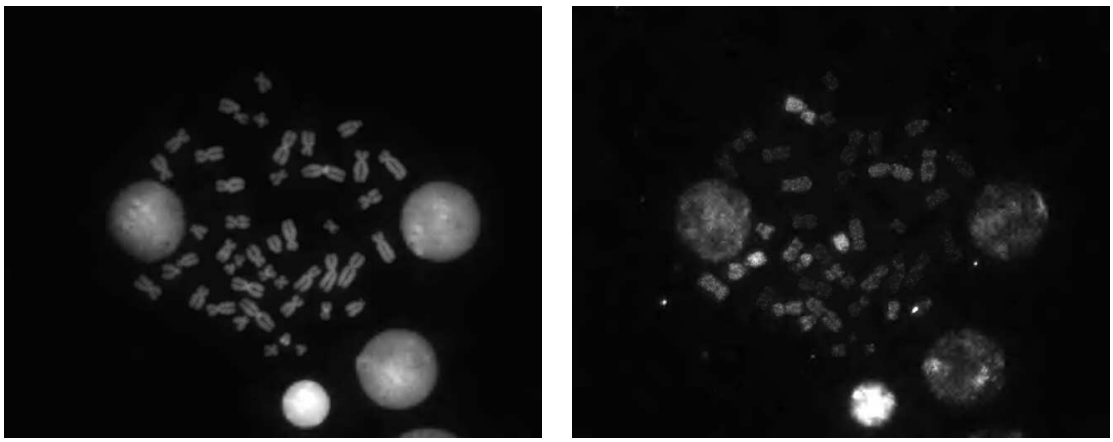


(b)

Fig. 8. Two channels of M-FISH image set "A0101XY" reconstructed at different bit-rates. The images in the left column are from the DAPI channel and the right from the Texas Red channel. (a) The original images. (b) Reconstructed at 0.01 bits/pixel/channel. (c) Reconstructed at 0.025 bits/pixel/channel. (d) Reconstructed at 0.05 bits/pixel/channel. (e) Reconstructed at 0.1 bits/pixel/channel. (f) Reconstructed at 0.15 bits/pixel/channel. The bit-rates referred are for coding the background only.

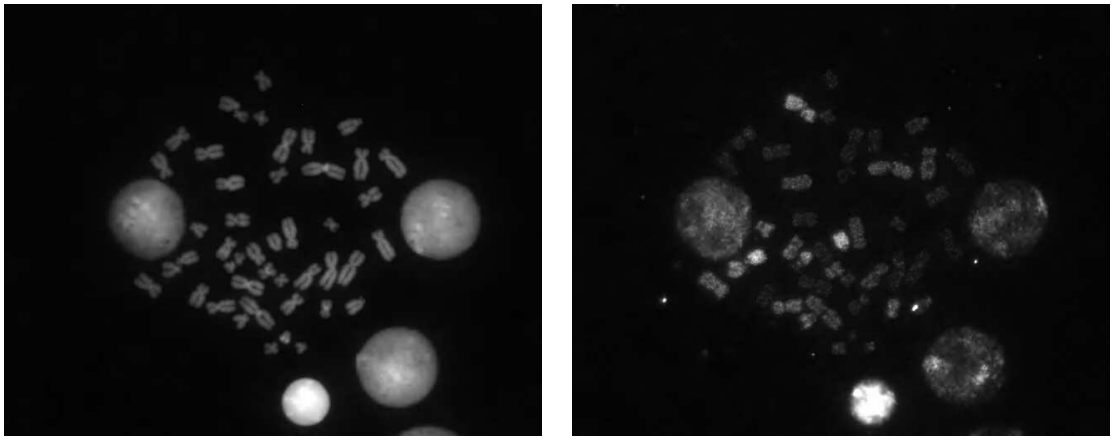


(c)

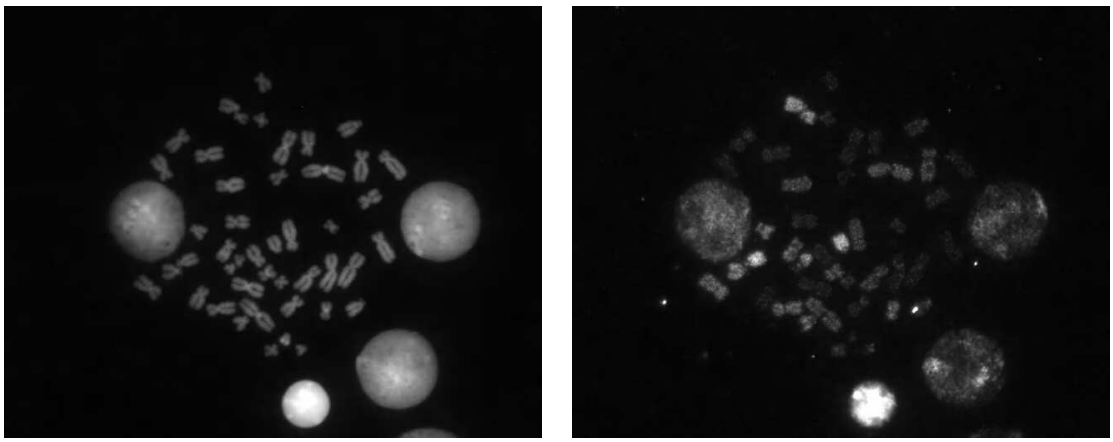


(d)

Fig. 8 continued.



(e)



(f)

Fig. 8 continued.

CHAPTER III

MICROARRAY IMAGE PROCESSING*

This chapter addresses two important issues associated with microarray image processing, signal estimation and image compression, by introducing a new integrated scheme: Microarray BASICA.

A. Overview of Micorarray Image Processing

As explained in the first chapter, microarray images cannot be used for genomic data analysis directly. Appropriate image processing procedures are to be performed in order to estimate the expression levels from the images for downstream analysis. Thousands of cDNA target sites must first be identified as the foreground by an image segmentation algorithm. Then the intensity pair (R, G) that represents gene expression levels of both channels is estimated from every foreground target site with appropriate background adjustment. Subsequent data analysis is usually conducted based on the log-ratio $\log R/G$ of the intensity pair. As the very first step of cDNA microarray signal processing, the accuracy of signal estimation is critical to the reliability of subsequent data analysis. Many signal estimation schemes have been developed for this purpose in recent years and can be found in various commercial and non-commercial software packages [49]-[65].

Besides signal estimation, image compression also raises notable attentions. Generally, because each channel of the microarray image is typically more than 15 MB in size, highly efficient compression is necessary for data backup and communication

*Reprinted from EURASIP Journal on Applied Signal Processing, vol. 4, J. Hua, Z. Liu, Z. Xiong, Q. Wu, and K. R. Castleman, "Microarray BASICA: Background adjustment, segmentation, image compression and analysis of microarray images," pp. 92-107, Copyright(2004) with permission from HINDAWI Publishing Corp.

purposes. Even with limit samples, the total space need to store these images can easily surpasses 1GB. In order to save storage space and alleviate the transmission burden for data sharing, the search for good progressive compression schemes that provide sufficiently accurate genetic information for data analysis at low bit-rates, while still ensuring good lossless compression performance, has become the focus of cDNA microarray image compression research recently [49, 50, 66].

B. Details of Microarray BASICA

In this chapter we introduce a new integrated system called Microarray BASICA. Microarray BASICA provides solutions to both signal estimation and image compression of cDNA microarray images. The major components of BASICA and their relationship with the elements of a microarray experiment are shown in Fig. 9. The upper two blocks, i.e., **Segmentation** and **Background adjustment** perform the signal estimation function, while the lower two blocks, i.e., **Header file** and **Compression** finish the image compression function.

Each two-channel microarray image acquired through the laser scanner is first sent to the **Segmentation** component, where the target sites are identified. With the result of segmentation, the **Background adjustment** component estimates each spot's foreground and background intensities, and calculates the log-ratio values based on the background-subtracted intensities. After this, the calculated log-ratio values along with the segmentation information and other necessary data related to each spot are output for downstream data analysis. In the mean time, BASICA compiles the segmentation result and estimated intensities into a header file. With this header file, the **Compression** component encodes the foreground and background of both channels of the original image into progressive bitstreams separately. The generated

bitstreams, plus the header file, are saved into a data archive for future access, or transmitted as shared data. On the other hand, to utilize the archived or transmitted data, BASICA can either quickly retrieve the necessary genetic information saved in the header file, or reconstruct the microarray image with available bitstreams through the **Reconstruction** component, and redo the segmentation and background adjustment.

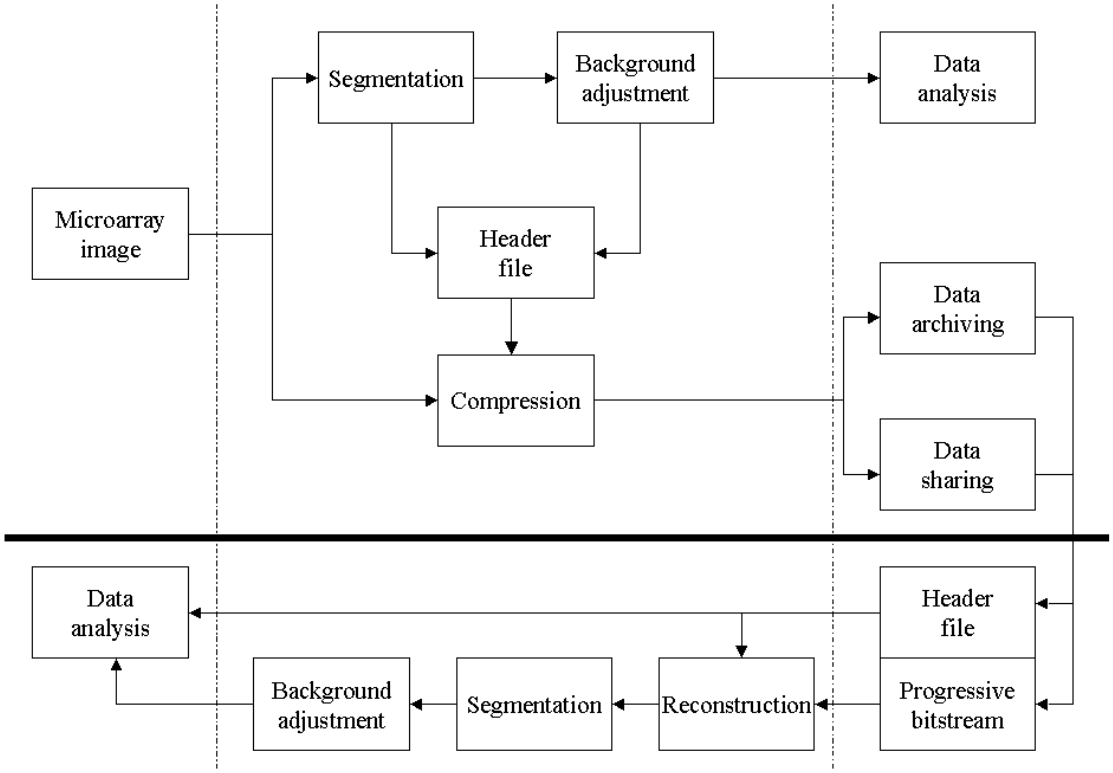


Fig. 9. The major units of BASICA.

1. Signal Estimation

Segmentation is performed to identify the target sites in each spot where the hybridization occurs. In [54], various existing segmentation schemes are summarized

and categorized into four groups: (1) fixed circle segmentation, (2) adaptive circle segmentation, (3) adaptive shape segmentation, and (4) histogram-based segmentation.

Although the shape of a target site is determined by the physical attributes of the DNA probes and the mechanism of the printing procedure, most target sites are round or donut-like in shape. The fixed circle segmentation, which sets a round region of constant diameter in the middle of each spot as the target site, appears to be the most straightforward method and is provided in most existing software packages [55, 57, 58, 59, 63, 64]. The radius of the foreground is set either by a default value as a parameter of the robot arrayer and laser scanner, or empirically determined by the user. The fixed circle method runs fast and performs well when the microarray spots are perfectly hybridized and aligned. In practical cases, however, the spots are far from perfect due to unpredictable non-uniform hybridization across the spot or misalignment of the probe array. Genepix [59] uses the adaptive circle segmentation to accommodate the varying sizes of different target sites, and Dapple [57] finds the best matched position of the round region in each spot to cope with the misalignment.

Neither the fixed nor the adaptive circle segmentation can accommodate the irregular shapes of the target sites in the images. To tackle this problem, more accurate and sophisticated segmentation methods are needed. The segmentation technique introduced in [54] uses *seeded region growing* [67], while other methods [51, 52, 56, 61, 63, 65] rely on more conventional histogram-based segmentation algorithms. The histogram-based methods generally compute a histogram of pixel intensities for each spot. Methods in [56, 63, 65] adopt a percentile-based approach, which sets the pixels in a high percentile range of the histogram as the foreground and those in a low range as the background. Methods in [52, 61] use a threshold-based approach. To ensure correct segmentation, methods in [56, 61] employ repetitions to

find the most stable segmentation. The histogram-based segmentation demonstrate good performance when a target site has a high hybridization rate, i.e. a high intensity. However, the intensities of most target sites are actually very close to the local background intensities, and it is hard to segment correctly by finding a threshold based on the histogram only. In an attempt to solve this problem, Chen *et al.* introduced a Mann-Whitey-test-based segmentation method in [51].

So far no single segmentation algorithm can meet the demands of all microarray images. Segmentation algorithms are normally designed to perform well on microarray images acquired by certain type of arrayers and scanners. It is therefore hard to compare them directly.

a. Mann-Whitney-test-based Segmentation

In BASICA, we use the Mann-Whitney-test-based segmentation algorithm introduced by Chen et al. in [51]. The Mann-Whitney test is a distribution-free rank-based two-sample test, which can be applied to various intensity distributions caused by irregular hybridization processes that are difficult to handle by conventional thresholding methods. Here we first give a brief description of the Mann-Whitney-test-based segmentation algorithm.

Consider two independent sample sets X and Y. Samples X_1, X_2,...,X_m are randomly selected from set X, and Y_1, Y_2,...,Y_n are randomly selected from set Y. All $N = m + n$ samples are sorted and ranked. Denote R_i as the rank of the i-th sample, $R(X_i)$ as the rank of sample X_i, and $R(Y_i)$ as the rank of Y_i. These ranks are used to test the following hypotheses:

$$\begin{aligned} H_0 &: P(X < Y) \geq 0.5 \\ H_1 &: P(X < Y) < 0.5. \end{aligned} \tag{3.1}$$

Define the rank sum of the m samples from X as

$$T = \sum_{i=1}^{m} R(X_i). \tag{3.2}$$

To avoid deviations caused by ties, T is commonly normalized as:

$$\overline{T} = \frac{T - m\frac{N+1}{2}}{\sqrt{\frac{nm}{N(N-1)}\sum_{i=1}^{N} R_i^2 - \frac{nm(N+1)^2}{4(N-1)}}}. \tag{3.3}$$

Hypothesis H_0 will be rejected if $\overline{T}$ is greater than a certain quantile $w_{1-\alpha}$, where α is the significance level.

In microarray image segmentation, hypothesis H_1 corresponds to the case that the intensities of the pixels in the foreground X is higher than the intensities of the pixels in the background Y, and hypothesis H_0 corresponds to the otherwise case. To segment a target spot, a predefined target mask (obtained by selecting, unifying and thresholding strong targets) is first applied to the spot. Pixels inside the mask correspond to set X, and pixels outside correspond to set Y. To start the test, n samples are randomly selected from set Y, while m samples with lowest intensities are selected from set X. If the hypothesis H_0 is accepted, the pixel with lowest intensity is removed from set X and m sample pixels are reselected. The test is repeated until hypothesis H_0 is rejected. Then the pixels left in set X are considered as the foreground at significance level α. The foregrounds obtained from the two channels are united into one to produce the final segmentation result.

The repetitive nature of this algorithm makes it cumbersome for real-time implementation. So in BASICA we proposed a fast Mann-Whitney-test-based algorithm [49] which runs much faster while generating identical segmentation results.

b. Speeding Up Mann-Whitey-test-based Segmentation Algorithm

Assume the predefined target mask is obtained according to the way described in [51, 52]. $X_1, X_2, ..., X_m$ and $Y_1, Y_2, ..., Y_n$ are picked from the foreground and background respectively. Without loss of generality, it suffices to assume $X_1 \leq X_2 \leq ... \leq X_m$ and $Y_1 \leq Y_2 \leq ... \leq Y_n$. Since $X_1, X_2, ..., X_m$ are m smallest samples in set X, all other samples can be determined if X_1 is set. Then Mann-Whitney-test-based segmentation is actually an optimization problem of minimizing X_1 subject to $\overline{T} \geq w_{1-\alpha}$. Chen et al.'s approach takes a large number of repetitions to reach the final segmentation. However, it turns out that the number of repetitions can be significantly reduced by carefully choosing the starting point and search strategy.

BASICA first finds an upper bound of the optimal X_1, denoted as X_1^{max}, which is related to $Y_1, Y_2, ..., Y_n$. With Eq. (3.3), $\overline{T} \geq w_{1-\alpha}$ can be written as

$$\sum_{i=1}^{m} R(X_i) \geq w_{1-\alpha} \sqrt{\frac{nm}{N(N-1)} \sum_{i=1}^{N} R_i^2 - \frac{nm(N+1)^2}{4(N-1)}} + m\frac{N+1}{2}. \tag{3.4}$$

In the right hand side of Eq.(3.4), only $\sum_{i=1}^{N} R_i^2$ is associated with X_1. If no tie exists, the ranks are from 1 to N and the sum is $\sum_{i=1}^{N} i^2$. If there is a tie, the ranks of the tied samples are the average of those ranks if there would have been no tie, and induce a reduction on the sum. A property of this reduction is that it is only related to the number of samples tied at that value. If there are k samples having the same value, the deduction is $\frac{1}{12}(k^3 - k)$. With this property, one can easily reduce the upper bound of $\sum_{i=1}^{N} R_i^2$. Assume ΔY is the decrease in the sum caused by the ties in sorted Y_1, Y_2,...,Y_n, then we have

$$\sum_{i=1}^{N} R_i^2 \leq \sum_{i=1}^{N} i^2 - \Delta Y, \tag{3.5}$$

where the equation holds when X_1, X_2,...,X_m have no tie among themselves and share

no tie with any sample in Y_1, Y_2,...,Y_n. In most cases the difference is very small and the bound is quite tight.

To simplify the notation, $\sqrt{\frac{nm}{N(N-1)}(\sum_{i=1}^{N} i^2 - \Delta Y) - \frac{nm(N+1)^2}{4(N-1)}}$ in Eq. (3.4) is notated as σ_{max} in the rest of this chapter. Then X_1^{max} must satisfy the inequality

$$\sum_{i=1}^{m} R(X_i) \geq w_{1-\alpha}\sigma_{max} + m\frac{N+1}{2} \tag{3.6}$$

no matter what $X_2, X_3, ..., X_m$ can be for as long as the assumption $X_1 \leq X_2 \leq ... \leq X_m$ holds. So to find X_1^{max} is to find the smallest X_1 that the smallest rank sum of $X_1 \leq X_2 \leq ... \leq X_m$ still satisfies inequality (3.6). To associate X_1^{max} with known information $Y_1, Y_2, ..., Y_n$, assuming $Y_u < X_1^{max}$. Then the minimum rank sum is $\sum_{i=1}^{m} R(X_i) = \sum_{i=1}^{m}(u+i)$, when $Y_u < X_1 \leq X_2 \leq ... \leq X_m < Y_{u+1}$. By solving the inequality with $\sum_{i=1}^{m} R(X_i) = \sum_{i=1}^{m}(u+i)$, u can be obtained as:

$$u = \lceil \frac{w_{1-\alpha}\sigma_{max}}{m} + \frac{n}{2} \rceil. \tag{3.7}$$

Thus the upper bound X_1^{max} is the smallest sample in X that is larger than Y_u. For any sample set $X_1, X_2, ..., X_m$ with $X_1 \geq X_1^{max}$, hypothesis H_0 can be rejected outright. The threshold $\min X_1$ subject to $\overline{T} \geq w_{1-\alpha}$ must be smaller than X_1^{max} and can be checked out by perform the Mann-Whitney test repetition backwardly. Since $X_1, X_2, ..., X_m$ normally have similar intensities which bring on consecutive ranks, X_1^{max} is usually very close to the actual threshold. Hence the repetitions can be greatly reduced if backward repetitions based on X_1^{max} are applied.

Besides changing the starting point and repetition direction, a two-tier repetition strategy can be used to reduce the repetition in case when the upper bound is not so tight as expected. In the first tier, one does not perform the repetition in a pixel-by-pixel manner, but in a leaping manner instead. Then a pixel-by-pixel repetition

follows up and locates the exact segmentation in the second tier. Larger step size means fewer repetitions in the first tier but more in the second tier, while smaller step size has the opposite effect. A natural choice of the repetition steps is indicated by Y_1, Y_2,...,Y_n when n is not very large. The whole algorithm is described as follows:

- Step 1: Calculate u using Eq. (3.7);
- Step 2: Find m smallest samples from set X that are larger than Y_u, and execute the Mann-Whitney test;
- Step 3: If hypothesis H_0 is rejected, then set $u = u - 1$ and go to step 2, otherwise, go to step 4;
- Step 4: $u = u + 1$. Find m smallest samples from set X that are larger than Y_u, and begin the pixel-by-pixel repetition in backward manner.

It should be noted that this modified Mann-Whitney-test-based segmentation algorithm may not always generate identical results with Chen et al.'s original algorithm. In order to obtain identical results, the backward- searching nature of the new algorithm requires the normalized rank sum in Eq. (3.3) to be strictly increasing during the repetition of the original algorithm. This is not guaranteed due to the occurrences of ties in the sorted samples. In one extreme case, when all N samples have the same intensity, the devisor will become zero and the normalized rank sum will be infinity. Actually Chen et al.'s original algorithm can be viewed as trying to find the largest foreground that rejects the hypothesis H_0, while the modified algorithm in BASICA tries to find the smallest foreground that accepts the hypothesis H_0. Since in most cases the normalized rank sum will be strictly increasing, we expect the segmentation results of the modified algorithm to be identical to the original algorithm most of the time.

The comparisons of the number of required repetitions between Chen et al.'s algorithm and our modified algorithm are given in Table V. We find that the segmentation results on all test spots of the sample images used in this study are identical between the original algorithm and the modified algorithm. From the table we observe that the modified algorithm reduces the number of repetitions by up to 50 times from what is required of the original algorithm.

Table V. The comparisons on the number of repetitions between Chen et al.'s algorithm and our modified method used in BASICA at different significance levels. Results are averaged over 504 spots in both channels from different test images. Both algorithms set $m = n = 8$ and use the same randomly selected samples from the predefined background for the Mann-Whitney test.

α	0.001	0.005	0.01	0.05
Chen et al.	328.7	269.1	270.9	226.3
BASICA	7.5	7.3	5.9	3.7

c. Post Processing

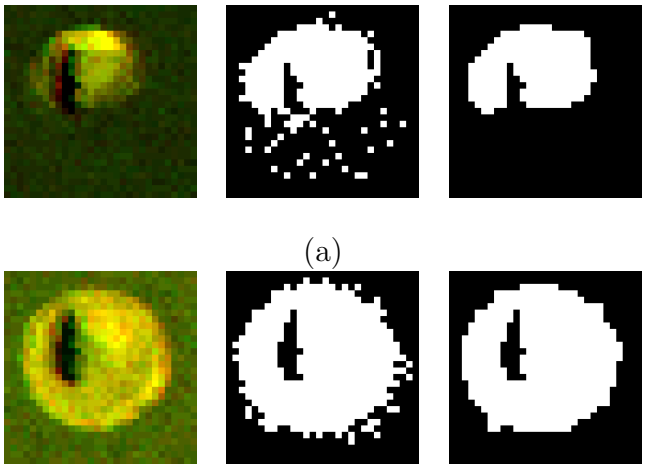
Like common threshold-based segmentation algorithms, there are always many annoying shape irregularities in the segmentation results obtained by the Mann-Whitney-test-based algorithms. These irregularities occur randomly and can severely reduce the compression efficiency. Thus an appropriate post-processing procedure is necessary to achieve efficient compression. Moreover, because most irregularities are pixels with a high probability of noise corruption, eliminating them is unlikely to compromise the accuracy of subsequent data analysis.

In BASICA, we categorize possible irregularities into two types and employ different methods to eliminate them. The first type includes isolated noisy pixels or

tiny regions, which can be observed from the lower half of the segmentation result in Fig. 10 (a). These irregularities are caused usually by nonspecific hybridization or undesired binding of fluorescent dyes to the glass surface. The second type includes the small branches attached to the large consolidated foreground regions, which are visible in the segmentation results of Fig. 10 (a). Located between the foreground and background, intensities of these irregularities are also in-between, making them vulnerable to noise corruption. The irregularities in most segmentation results are usually made up of both these two types. For the first type, BASICA will detect and remove them directly from the foreground. As to the second type, BASICA applies an operation similar to the standard morphological pruning [68]. By removing and pruning repetitively, BASICA can successfully eliminate most irregularities in three to five repetitions. The right column of Fig. 10 shows the post-processing results on the original segmentation, which are to be used for the compression of the images. Fig. 11 shows a portion of a microarray image and its segmentation results.

d. Background Adjustment

It is commonly believed that the pixel intensity of the foreground reflects the joint effects of the fluorescence and the glass surface. To obtain the expression level accurately, the intensity bias caused by the glass surface should be estimated and subtracted from the foreground intensity, and this process is known as background adjustment. Since there is no hybridization in the background area, the background intensity is normally measured and treated as an intensity bias. Although mean pixel intensity has been adopted in almost all existing schemes as the foreground intensity, several methods have been developed for background intensity estimation. The major differences of various methods lie in two aspects: (1) on which pixels the estimation is based, and (2) how to calculate the estimation. Regarding the first aspect,

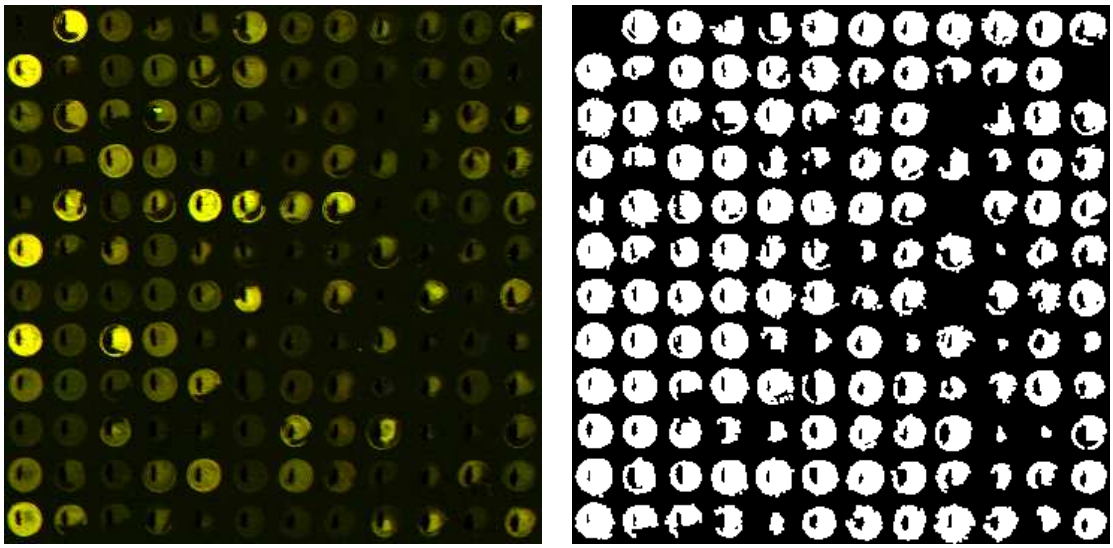


(a)

(b)

Fig. 10. Segmentation and post-processing of two typical spots. The left column shows the original microarray spots in RGB composite format. Some intensity adjustments are applied in order to show them clearly. The middle column shows the corresponding segmentation results using the Mann-Whitney test with significance level $\alpha = 0.001$. The right column shows the final segmentation results after post-processing.

the regions chosen for background estimation vary from a global background to a local background. For the global background, the background regions in all spots are considered, and a global background intensity is estimated and subtracted from every foreground intensity [55, 62]. The global background ignores possible variance between sub-arrays and spots. So in [55] partial global background estimation is performed based on the background of one sub-array or on several manually selected spots. The more common approach is to estimate the background intensity based on the local background for each target site separately. The local background can be the entire background region in one spot [64], or, to avoid interference from the foreground, it can be the region with a certain distance from the foreground target site [53, 57, 59, 61, 63]. In the extreme case, the algorithm in [60] used the pixels on the border of each spot as the local background. However, using too few pixels increases



(a) (b)

Fig. 11. (a) Part of a typical cDNA microarray image in RGB composite format. Some intensity adjustments were applied in order to show the image clearly. (b) The segmentation results of (a).

the possibility of a large variance in background estimation. As to the second aspect, almost all existing systems adopt mean or median to measure the expression level. Besides these, mode and minimum are also used in some softwares [52, 62]. Unlike all the methods mentioned above, a morphological opening operation is performed in [54] to smooth the whole background and then estimate the background by sampling at the center of the spot.

Some commercial software packages [55, 62] offer more than one choice for background adjustment. ArrayMetrix [55] provides up to nine methods, while ArrayVision [56] provides seven ways of background region determination and six choices of averaging method. Experiments in [54] show that the selection of the background adjustment methods has significant impact on the log-ratio values subsequently obtained. However, there is no known criterion to measure whether a certain approach is more accurate over the others.

BASICA chooses the average of pixel intensities in the local background as the estimate of background intensity. To prevent possible biases caused by either the higher intensity values of the pixels adjacent to the foreground target sites or the lower intensity values of the dark hole regions in the middle of the spots, the local background used in BASICA is the background defined by the predefined target mask obtained through the segmentation.

2. Image Compression

In order to achieve the best lossy-to-lossless compression performance, a novel distortion measure is introduced to match the requirement of data analysis. Thus in this section we first provide the background knowledge on low-level statistical data analysis, then introduce our microarray image compression scheme.

a. Data Analysis

Because so many elements impact the pixel intensities of the microarray image, genetic researchers do not use the absolute intensities of the two channels, but the ratio of them to measure the relative abundance of gene transcription. Not all genetic information estimated are reliable enough for data analysis. If the spot has so poor quality that no reliable information can be estimated, it is qualified as a false spot, otherwise, it is a valid spot. For a valid spot k, the expression ratio is denoted as

$$T_k = \frac{R_k}{G_k} = \frac{\mu_{FR_k} - \mu_{BR_k}}{\mu_{FG_k} - \mu_{BG_k}}, \tag{3.8}$$

where R_k and G_k are the background-subtracted mean intensities of the red and green channels respectively, μ_{FR_k} and μ_{FG_k} are the respective foreground mean intensities, and μ_{BR_k} and μ_{BG_k} are the respective estimated background mean intensities. Because expression ratio has an unsymmetric distribution, which contradicts the basic

assumptions of most statistic tests, the log-ratio $\log T_k = \log R_k/G_k$ is commonly used instead in most applications. In addition to the log-ratio, an auxiliary measure which is often helpful to data analysis is the log-product $\log R_k G_k$. However, since the log transform does not have constant variance at different expression levels, some alternative transforms like glog [69] have recently been introduced. In gene expression studies, such transformed ratios are ordinarily normalized and quantized into three classes: down-regulated, up-regulated and invariant. Expression level estimation and quantization provide the starting point for subsequent high-level data analysis and their accuracy is crucially important. Therefore compression schemes should be designed to minimize the distortion in the image, and their performance should be assessed by agreement/disagreement in gene expression level measurement caused by the compression. These topics will be discussed in detail in the later sections.

b. Image Compression

Since microarray images contain huge amounts of data and are usually stored at the resolution of 16 bpp, a two-channel microarray image is typically between 32 and 64 megabytes in size. Efficient compression methods are highly desired to accommodate the rapid growth of microarray images and to reduce the storage and transmission costs. Currently the common method to archive microarray images is to store them losslessly in TIFF format with LZW compression [8, 9]. However, such an approach does not exploit 2-D correlation of data between pixels and does not support lossy compression. Due to the huge data size, microarray images require efficient compression algorithms which support not only lossless compression but also lossy compression with graceful degradation of image quality for downstream data analysis at low bit-rates.

Recently a new method known as the segmented LOCO (SLOCO) was intro-

duced in [66]. This method exploits the possibility of lossy-to-lossless compression for microarray images. SLOCO is based on the LOCO-I algorithm [70], which has been incorporated in the lossless/near-lossless compression standard of JPEG-LS. SLOCO employs a two-tier coding structure. It first encodes mircoarray images lossily with near-lossless compression, then applies bit-plane coding to the quantization error to refine the coding results until lossless compression is achieved. SLOCO can generate a partially progressive bitstream with a minimum bit-rate determined by the compression of the first tier, and the coding is conducted on the foreground and background separately.

In BASICA we also incorporate lossy-to-lossless compression of microarray images. The aims of compression in BASCIA are twofold: 1) To generate progressive bitstreams that can fulfill the requirements of signal processing and data analysis at low bit-rates for data sharing and transmission applications and 2) to deliver competitive lossless compression performance for data archiving applications with a progressive bitstream. To achieve these objectives, the compression scheme in BASICA treats the foreground and background of microarray images separatively. Obviously the foreground and background usually have significant intensity differences and they are relatively homogeneous in their corresponding local regions. Hence by compressing the foreground and background separately, the compression efficiency is expected to improve significantly. This is done by utilizing the outcomes of segmentation. Before encoding, BASICA saves all necessary segmentation information into a header file for subsequent compression.

SLOCO in [66] is based on spatial-domain predictive coding. In contrast, BASICA employs bit-plane coding in the transform domain. Bit-plane coding enables BASICA to achieve truly progressive bitstream coding at any rates. To allow lossy compression, an appropriate distortion measurement is needed. Generally, medical

image compression requires visually imperceptible differences between the lossily reconstructed image and the original. Traditional distortion measures, such as mean square error (MSE), are poor indicators for this purpose. However, unlike other types of medical images, the performance of microarray image compression does not depend on visual quality judgement, but instead on the accuracy of final data analysis. Therefore it is reasonable to adopt a distortion measure adherent to the requirements of data analysis. Since almost all existing data analysis methods use the transformed expression values, we should seek to minimize the distortion under these measurements. In BASICA we adopt distortion measures based on the log-ratios and the log-products because they are the mostly used transforms in common applications. However, as we will see later, the scheme employed in BASICA can be easily adapted for other transform measures.

The log-ratios and the log-products decouple the data of two channels into two separate log-intensities, $\log R$ and $\log G$. This ensures that the compression can be done on each channel independently. Without loss of generality, we only refer to the R channel in the rest of the paper.

BASICA currently employs the MSE of $\log R$ as the distortion measurement, which is defined as

$$MSE_{\log R} = \frac{1}{N}\sum_{i=1}^{N}(\log R_i - \log \hat{R}_i)^2, \tag{3.9}$$

where N is the total number of spots in the microarray image, and R_i and $\hat{R}_i$ are background subtracted mean intensities obtained from spot i of the original and reconstructed image respectively.

There is a direct relationship between the MSE of log-intensity and the traditional

MSE. For spot k, its log-intensity $\log R_k$ can be further written as

$$\log R_k = \log(\mu_{FR_k} - \mu_{BR_k}) = \log(\frac{1}{M_k}\sum_{i=1}^{M_k} X_i - \mu_{BR_k}), \tag{3.10}$$

where M_k is the total number of pixels in the foreground of spot k, and X_i is the intensity of the i-th pixel. So the unit error $\Delta \log R_k$ is associated with the unit error ΔX_j of j-th pixel by

$$\Delta \log R_k = \frac{\Delta X_j}{M_k(\mu_{FR_k} - \mu_{BR_k})}. \tag{3.11}$$

For the pixels in the background, because most existing schemes do not compute the average intensity as μ_{BR_k} but use non-linear operations such as modulo or median filtering, the above derivation no longer holds. The foreground and background pixels have different impacts on the log-intensity and should be considered separately.

Eq. (3.11) indicates that the MSE of log-intensity is actually a weighted version of traditional MSE. The weight $\frac{1}{M_k(\mu_{FR_k}-\mu_{BR_k})}$ is a constant for pixels in the same spot and is inversely proportional to the spot's intensity and foreground size. The higher a spot's intensity or foreground size, the larger its allowable reconstruction error.

Quite similarly, one can easily derive other MSE distortion measurements for other transforms. For example, the glog transform in [69] is

$$g(R_k) = \log(\mu_{FR_k} - \alpha + \sqrt{(\mu_{FR_k} - \alpha)^2 + c}), \tag{3.12}$$

where α and c are parameters estimated from the microarray image. Then with straightforward derivation, one can associated the unit error $\Delta g(R_k)$ with the unit error ΔX_j of j-th pixel by

$$\Delta g(R_k) = \frac{\Delta X_j}{M_k\sqrt{(\mu_{FR_k} - \alpha)^2 + c}}. \tag{3.13}$$

Thus the MSE of glog is also a weighted version of traditional MSE, and like MSE of

log-ratio, the measurement allows larger distortions in spots of high intensities.

Although we can derive different distortion measurements for different transform, the compression scheme in BASICA can only be designed based on one type of distortion measurement. As mentioned before, in BASICA we choose MSE of log-ratio as the distortion measurement.

With the help of Eq. (3.11), we introduce a new lossy-to-lossless compression scheme in BASICA by modifying EBCOT [10] with several techniques specifically designed for the requirements of microarray technology. First, like what have been done in the M-FISH image compression at previous chapter, we modify the EBCOT to compress arbitrarily shaped regions by applying *critically sampled* wavelet transform, and encoding the foreground and background separately. Then we apply intensity shifts and bit shifts on the coefficients to minimize the MSE of log-intensity.

Since EBCOT, which is a state-of-the-art compression algorithm incorporated in JPEG-2000 standard, is the basis of EWV mentioned in pervious chapter, here we will omit those unnecessary description on EBCOT, and focus on our major modifications to EBCOT:

- **Header file:** A header file is necessary for saving the information which will be used in the encoding and decoding procedures. To ensure that the encoder and decoder can correctly compress and reconstruct the foreground and background independently, the segmentation information must be saved in the header file. Besides, Eq. (3.11) indicates that the mean intensities of the foreground and background are also needed by the compression algorithm. To save storage memory, these data are coded with LZW compression. Although the segmentation information and spot intensities are enough for the compression component, other data, such as variances of pixel intensities in each spot, can also be saved

in the header file for quick genetic information retrieval. In the practical implementation, the header file will be generated before encoding and must be transmitted and decoded first.

- **Shape-adaptive integer wavelet transform and object-based EBCOT:** This is pretty much like what we have done with EMIC in M-FISH image coding. Shape-adaptive integer wavelet transforms and bit-plane coding are applied to the foreground and background independently.

- **Intensity shifts:** To minimize the initial MSE, the average intensity of the image is subtracted from each pixel before encoding and added back after decoding. Unlike 8-bit natural images, the foreground of a microarray image normally has an exponential intensity distribution. The exponential distribution property of the foreground makes the global average intensity subtraction less effective. However, the pixels in the foreground of any spot k normally have similar intensities and roughly have a symmetric distribution around μ_{FR_k}. So for the encoding of the foreground, instead of a global average intensity subtraction, each pixel in spot k is subtracted with μ_{FR_k}. Since μ_{FR_k} is already saved in the header file, intensity shifts do not cost any overhead. With intensity shifts, the distribution of foreground intensities are transformed into a symmetric shape with a high peak around zero. And for the background compression, through our experiments we find that the pixels in the background actually have a roughly symmetric intensity distribution, suggesting that the global average intensity subtraction will be appropriate.

- **Bit shifts:** EBCOT uses block-based bit-plane coding. In order to minimize the distortions at different rates, one must code the bit-planes of different spots according to their impacts on the MSE of log-intensity. One straightforward so-

lution is to scale the coefficients of each spot with the spot's weight, so bits at the same bit-plane of all spots have the same impacts on the MSE of log-intensity. However, because the weights are non-integer fractions, lossless compression cannot be ensured under such a scaling. Furthermore, although one can round them to the closest integer as an approximation, any scaler w will increase a coefficient's information by up to $\lceil log_2 w \rceil$ bits, which can lead to a very poor lossless compression performance. In BASICA, we apply the scaling by bit shifts which is a good approximation and meanwhile does not compromise the performance of lossless compression. For spot k, BASICA obtains

$$S_k = \lfloor \log_2(M_k(\mu_{FR_k} - \mu_{BR_k})) + 0.5 \rfloor. \tag{3.14}$$

Let $S^{max} = \max\{S_1, S_2, \ldots, S_N\}$. Then it scales the coefficients of spot k by upshifting them $S^{max} - S_k$ bits.

- **Background compression:** With careful consideration, bit shifts have not been applied in the background compression in BASICA for several reasons. First, since there exist different approaches to compute the background intensity, and the values obtained by these methods also vary a lot, it is unclear how to find a unique weight for each pixel like what BASICA has for foreground compression. Second, unlike isolated target sites in the foreground, the local background is normally connected to each other. Thus bit shifts will bring abrupt intensity changes along the borders of spots, which will in turn lower the compression efficiency significantly in lossless coding performance. Even though one can figure out the weights through a formula similar to Eq. (3.11) based on certain background extraction methods, there will be a significant tradeoff on lossless compression, which is about 0.8 bpp according to our experiments. So

in BASICA, we apply a global average intensity subtraction and no bit shifts on the background compression, i.e., the traditional MSE measure is used for rate-distortion optimization. Normally the pixel intensities in the background are located in a very small range, which means the background is pretty homogenous. Thus compression with traditional MSE measure should be able to represent the background with fairly small bit-rates.

To this end, the final code of a two-channel microarray image is composed of five different parts: a header file, and two bitstreams representing the foreground and background respectively from each channel.

C. Experimental Results and Discussion

Experiments have been conducted to test the image compression performance of BASICA with eight microarray images from two different sources. We used three test images from the National Institutes of Health (NIH). Each of these images contains eight sub-arrays arranged in 2-by-4 format. In each sub-array the spots are arranged in a 29-by-29 format. There are a total of 20184 spots in all three NIH images. In addition to these, we also tested on another set of five test images obtained from Spectral Genomics Inc. (SGI). Each of the SGI images contains eight sub-arrays arranged in 12-by-2 format, and in each sub-array the spots are arranged in a 16-by-6 format. These five SGI images contain a total of 9960 spots. The target sites in the NIH images exhibit noticeable irregular hybridization effect and have irregular brightness patterns across the spots. The intensities of these target sites span over a large range and vary considerably. The target sites in the SGI images appear to be hybridized more homogeneously, and many of them have nearly perfect circular shape.

In the experiments, for each two-channel image, the summed bit-rate of all the

bitstreams from both channels, plus the shape information, were reported in bit-per-pixel(bpp) format, which either represents the compression bit-rate or the reconstruction bit-rate, depending on the type of test performed. And the corresponding bit-rate of uncompressed original image is 32bpp. BASICA first segmented the image and generated the header file. The average overhead of the header file was 0.5 bpp for the NIH images and 0.24 bpp for the SGI images, based on the post-processed segmentation results. The header file overheads were smaller on the SGI images because of different settings of the microarray arrayers used to acquire the images: there were much fewer spots in each SGI image than those in each NIH image. After generating the header file, the foreground and background of each channel were compressed independently.

1. Comparisons of Wavelet Filters and Decomposition Levels

The framework of proposed compression scheme in BASICA does not specify which wavelet filters and how many wavelet decomposition levels to use. In order to find the optimal choice for microarray image compression, we compare the results generated with different wavelet filters and decomposition levels. All the results presented in this section are based on the NIH images unless stated otherwise.

Table VI lists the lossless coding results by BASICA using nine different wavelet filters with one-level wavelet decomposition. From these we found that the compression results vary only in a small range of about 0.07 bpp. Among all nine sets of filters, the 5/3 wavelet filters achieved the best result. This is probably because the 5/3 wavelet filters have relatively shorter filter lengths, and therefore fit better with the not-so-smooth nature and the small size of microarray target sites. Nevertheless as the discrepancies in the results were small, the choice of the wavelet filters appeared to be not critical to the system performance.

Table VI. Lossless compression results (in bpp) of BASICA using different integer wavelet filters with one-level wavelet decomposition. The results are averaged over the NIH images.

Wavelet filters	9/7-F	(2+2,2)	5/3	S+P	(4,2)	(2,4)	(4,4)	(6,2)	2/6
File size	13.99	14.01	13.97	14.03	14.01	13.97	13.99	14.04	14.00

Table VII lists the lossless coding results by BASICA with different wavelet decomposition levels. Only the best-performing 5/3 wavelet filters were evaluated in these tests. The performance appeared to get worse when decomposition level increased and compression with only one-level decomposition achieved the best result. This is partly due to the fact that although with more decompositions more data energy is be compacted into smaller subbands, it also introduces a higher model-adaptation cost to arithmetic coding in the newly generated subbands, which cancels out the gains. Similar to the comparison among the wavelet filters, the discrepancies of lossless compression performance using different decomposition levels are very small.

Table VII. Lossless compression results (in bpp) of BASICA using the 5/3 wavelet filters with different wavelet decomposition levels. The results are averaged over the NIH images.

	1 level	2 levels	3 levels	4 levels	5 levels
File size	13.97	14.00	14.01	14.02	14.02

To confirm this observation lossy compression tests were also performed to compare the performances based on the choices of wavelet decomposition level. To evaluate the effect of lossy compression on data analysis, the test images were first reconstructed at a target rate. Then the reconstructed images were processed and genetic information (i.e. log-ratio) was estimated and compared with the same information

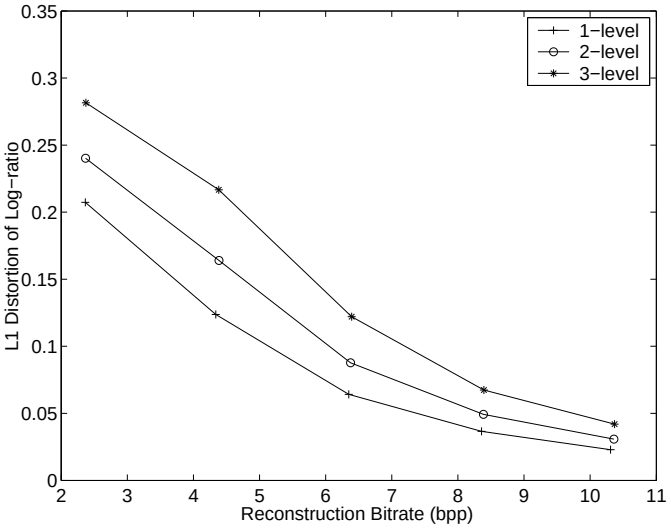
estimated from the original images. To ensure credibility of the comparisons, the Mann-Whitney-test-based segmentation started with the same selection of random pixels in the predefined background in both the reconstructed image and the original image. The segmentation was conducted under three different significance levels $\alpha = 0.001$, 0.01 and 0.05. At each significance level, log-ratios were estimated and distortions were computed. The distortions shown are the average distortions at three significance levels over the three test images. Both the l_1 distortion and l_2 distortion (i.e., MSE) of log-intensity were used as the error measures. Fig. 12 shows the average reconstruction errors using BASICA at different bit-rates with three different decomposition levels of the 5/3 wavelet transform.

From this figure we can see that one-level decomposition yielded a significantly better performance than the others. Based on the above lossless and lossy compression results, we decided to use the 5/3 wavelet filters with one-level wavelet decomposition as a default setting in BASICA.

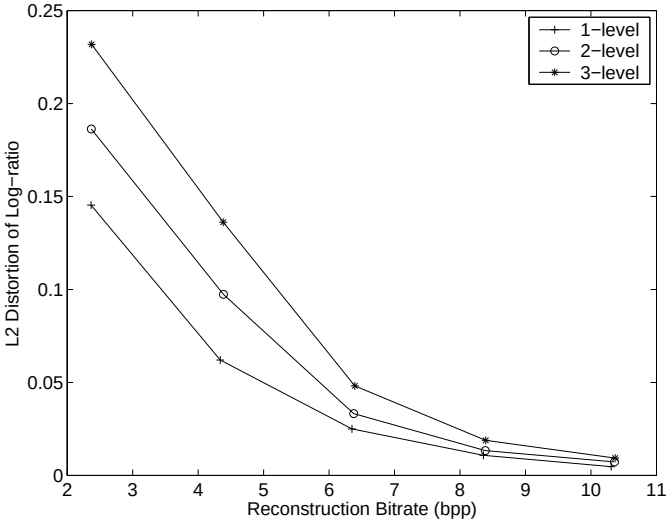
2. Comparisons of Lossless Compression

We first compared the lossless compression performance of BASICA with three current standard coding schemes: TIFF, JPEG-LS and JPEG-2000. In the comparisons, TIFF, JPEG-LS and JPEG-2000 all compress a microarray image as a single region and no header file is added. To evaluate the improvement brought by the post-processing in segmentation, along with the intensity and bit shifts in compression, we also performed the tests of BASICA without the intensity and bit shifts, and without post-processing respectively (denoted as BASICA w/o PP and BASICA w/o shifts respectively in the tables and figures below).

The coding results are shown in Table VIII. The TIFF format, which is commonly used in existing microarray image archiving systems, produced the poorest



(a)



(b)

Fig. 12. Rate-distortion curves of log-ratio in terms of (a) l_1 distortion and (b) l_2 distortion with different wavelet decomposition levels at different reconstruction bit-rates. 5/3 wavelet filters were used. The segmentation was performed at three different significance levels $\alpha = 0.001, 0.01$ and 0.05 and three log-ratios and their corresponding distortions were then obtained. The distortions shown are the averages of the three significance levels over the NIH images.

results: about 4 bpp worse than all the other methods compared. JPEG-LS achieved the best performance on the NIH images. But like TIFF, it does not support lossy compression. The proposed BASICA turned out to be about 0.27 bpp worse than JPEG-LS on the NIH images and 0.12 bpp better on the SGI images. Besides, BASICA was significantly better than JPEG-2000 with the savings of 0.48 bpp and 0.56 bpp on the NIH and SGI images respectively. BASICA without intensity and bit shifts yielded almost the same performance as BASICA in lossless compression. On the other hand, one can see clearly that the irregularities in segmentation reduced compression efficiency substantially. Without post-processing, the average size of a header file was 0.33 bpp larger than that of BASICA on the NIH images, and 0.09 bpp larger on the SGI images respectively. Thus BASICA with post-processing was preferred on all the test images.

Table VIII. Lossless compression results (in bpp) of different coding schemes.

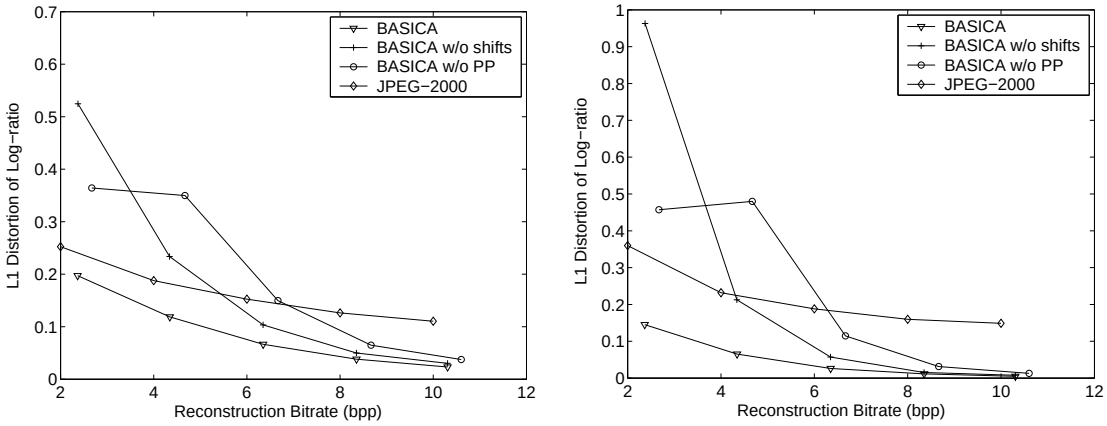
Methods	Bit-rates (NIH)	Bit-rates (SGI)
TIFF	18.27	17.21
JPEG-LS	13.70	14.49
JPEG-2000	14.45	14.93
BASICA w/o shifts	13.99	14.31
BASICA w/o PP	14.50	14.46
BASICA	13.97	14.37

3. Comparisons of Lossy Compression

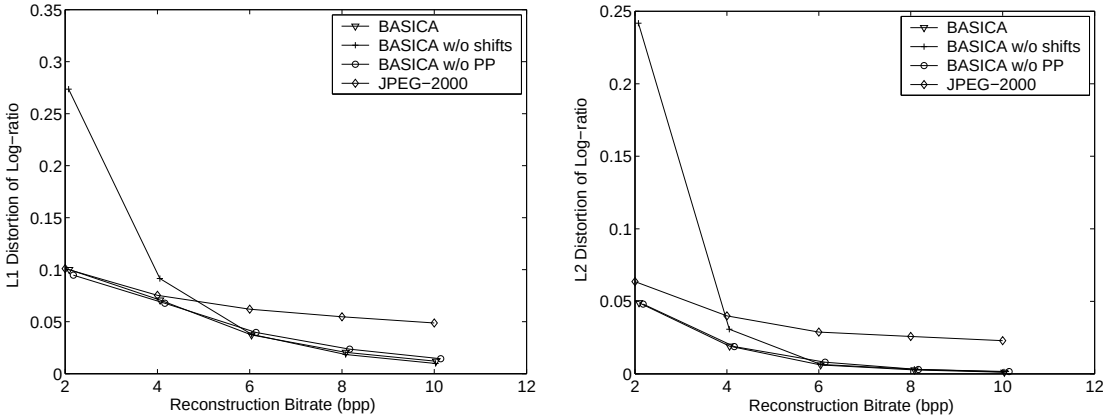
During the experiments, we also compared the lossy compression results at different bit-rates. Since TIFF and JPEG-LS do not support lossy compression functionality, JPEG-2000 was the only standard compression scheme compared in the experiments. Our comparisons were based on three different measurements.

a. Comparisons Based on l_1 and l_2 Distortions

We first compared the rate-distortion curves based on the l_1 distortion and l_2 distortion of log-intensity. Fig. 13 shows the average reconstruction errors of these methods at different bit-rates. We observe that, due to the effect of relatively more homogeneous hybridization, the distortion on the SGI images was uniformly smaller than the distortion on the NIH images. JPEG-2000 produced surprisingly small l_1 distortion values at low bit-rates, only inferior to BASICA on the NIH images and similar to the others on the SGI images. Nevertheless it produced relatively large l_2 distortion values. Apparently, without adjusting the MSE for log-intensity, JPEG-2000 spent too much bit-rate on high intensity pixels/spots, which led to high l_2 distortion. Furthermore, the distortion of JPEG-2000 decayed slowly in both l_1 and l_2 sense. For bit-rates beyond 6 bpp, it degraded to produce the worst distortion among all the methods. Without the intensity and bit shifts, BASICA performed poorly at lower bit-rates. Only when the bit-rate went above 6 bpp did its performance become acceptable. BASICA without post-processing produced different performances on images of different sources. On the NIH images, it obviously suffered from the irregularities of segmentation, yielding a performance between BASICA and BASICA without the intensity and bit shifts at low bit-rates. But it quickly became worse than both of these schemes when bit-rate increased. On the SGI images, in which target sites had more uniform hybridization, there was almost no difference between its performance and BASICA's. Compared to the other schemes, BASICA yielded the best performance in both l_1 and l_2 distortion at all bit-rates on all test images.



(a)



(b)

Fig. 13. Rate-distortion curves of log-ratio in terms of l_1 distortion (left column) and l_2 distortion (right column) under different reconstruction bit-rates for different compression schemes. (a) Results based on the NIH images. (b) Results based on the SGI images. The segmentation was performed at significance level $\alpha = 0.05$.

b. Comparisons Based on Scatter Plots

Besides l_1 and l_2 distortion measures, a more intuitively visual way to compare the distortion of different methods is by scatter-plotting. Fig. 14 shows the estimated log-ratios and log-products by different methods at a bit-rate around 4 bpp for two test images. In each scatter-plot, the blue diagonal line corresponds to the information estimated from the original images. From the plots we can see that BASICA had a better performance than the other methods. BASICA without post-processing had a worse performance on the NIH images and a good performance on the SGI images. JPEG-2000 and BASICA without intensity and bit shifts yielded worse performances on both sets of test images. This observation is consistent with the results shown in Fig. 13. Since a scatter-plot cannot provide quantitative performance measurements and can only visually display the data for comparisons at one bit-rate per plot, it does not provide a practical performance measurement.

c. Comparisons Based on Gene Expression Data

Rather than judging the performance based on the L1 and L2 distortion measures and the scatter plots, biologists and clinicians in gene expression studies are likely to care more whether a gene is differently detected or identified due to a lossy compression. Hence it is meaningful to look at rate of disagreement on detection and identification between lossily reconstructed image and original image. The detection and identification disagreement are defined as follows.

1) The detection disagreement is defined to be the valid spots in the original image being detected as false spot, or vice versa, after a lossy reconstruction.

2) The identification disagreement is defined to be a different classification outcome among up-, down-regulated, and invariant gene expression levels after a lossy

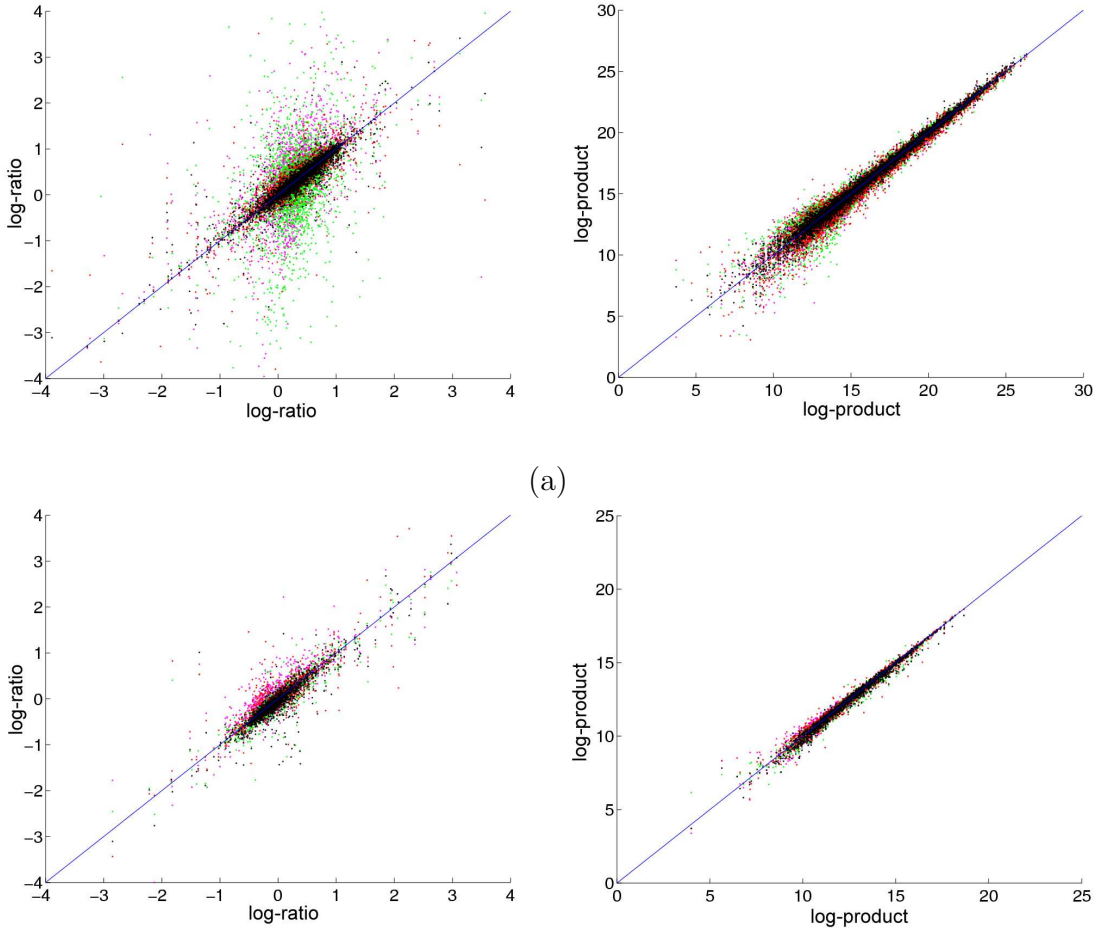
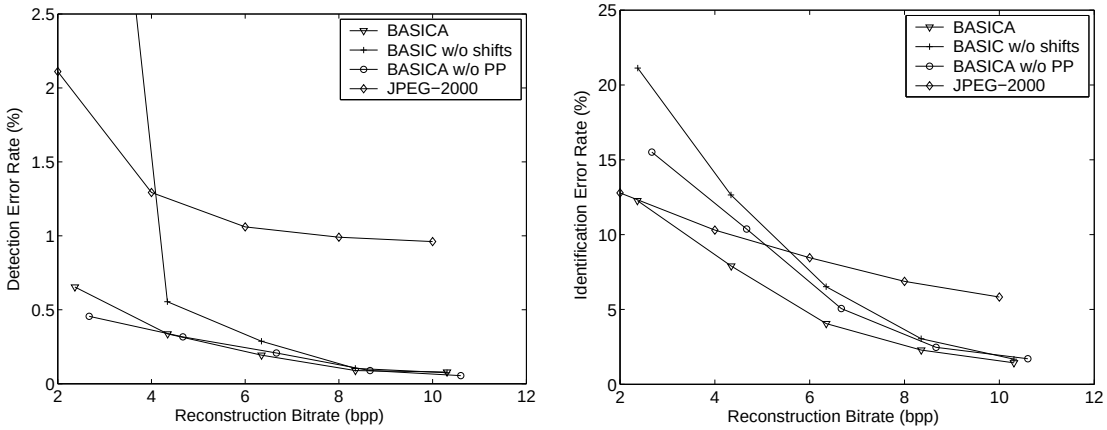


Fig. 14. Scatter-plots of log-ratio (left column) and log-product (right column) estimated from original images and reconstructed images using different schemes. (a) Results based on a NIH image. Black: BASICA at 4.3 bpp; Magenta: BASICA w/o shifts at 4.3 bpp; Green: BASICA w/o PP at 4.7 bpp; Red: JPEG-2000 at 4.0 bpp. (b) Results based on a SGI image. Black: BASICA at 4.1 bpp; Magenta: BASICA w/o shifts at 4.1 bpp; Green: BASICA w/o PP at 4.2 bpp; Red: JPEG-2000 at 4.0 bpp. The significance level in the Mann-Whitney test is $\alpha = 0.05$.

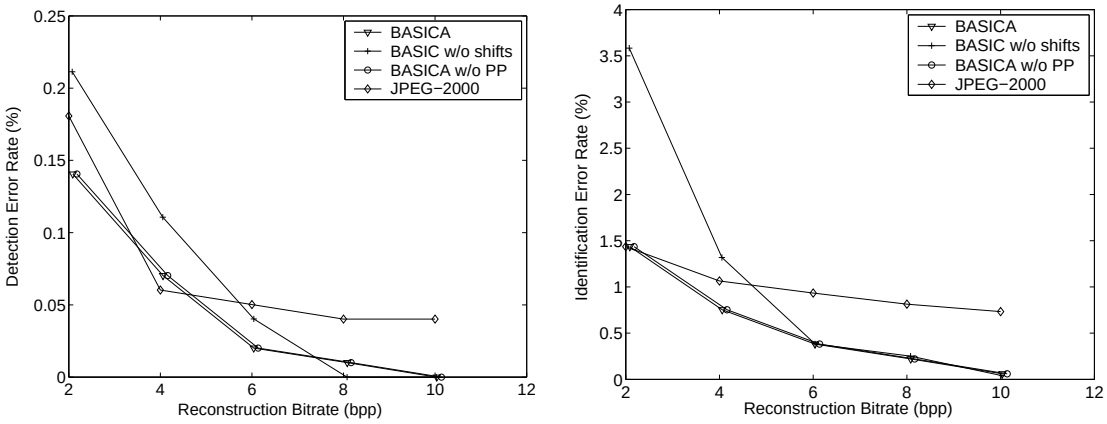
reconstruction, even though the detection outcome is same.

We conducted experiments using a simple quantitative model of gene expression data analysis to compare different methods. We determined that a spot was false if its foreground intensity was less than its background intensity in either channel, or no foreground target site was found by the segmentation. We also decided that if the log-ratio was larger or smaller than a certain threshold range $[\theta, -\theta]$, then the spot was up or down-regulated, otherwise it was invariant. For these experiments, no normalization was performed to reduce the inter-image data variations. The experiments were performed on the NIH images and the SGI images separately and the results are shown in Fig. 15. From this figure we can see that the identification disagreement rate was about 10 times higher than the detection disagreement rate. These results were similar to what have been shown in Fig. 13. The disagreement caused by the lossy compression of JPEG-2000 was comparable to that of BASICA only at 2 bpp, and dropped slowly when bit-rate increased. On the other hand, the disagreement caused by BASICA without intensity and bit shifts became acceptable only after 6 bpp. BASICA without post-processing yielded a performances similar to that of BASICA on the SGI images but did worse on the NIH images. One can also observe that the disagreement rates on the NIH images were much higher than on the SGI images at the same bit-rate. This is probably because NIH images are much noisier than SGI images, and hence require more bit-rates to compress. These results were consistent with Fig. 13, where the NIH images had much larger l_1 and l_2 distortion than the SGI images at the same bit-rate. For the NIH images, the identification disagreement rate was larger than 10% at 2 bpp and was around 1.5% at 10 bpp. For the SGI images, the identification disagreement rate was smaller than 2.5% even at 2 bpp, and was around 0.1% at 10 bpp. All these results consistently suggested that one could hardly find a common bit-rate that led to similar disagreement/agreement

rates for different microarray images. For images with homogeneous hybridization, which are becoming more available with the advance of microarray production technology, lossy compression at low bit-rates appears to be viable for highly accurate gene expression data analysis.



(a)



(b)

Fig. 15. The disagreement rates vs. the bit-rates. The threshold parameter $\theta = 1$. The segmentation was performed at significance level $\alpha = 0.05$. The left column plots depict the detection disagreement rates vs. the bit-rates. The right column plots depict the identification disagreement rates vs. the bit-rates. The disagreement rates shown are the averages of all images. (a) Results based on the NIH images. (b) Results based on the SGI images.

CHAPTER IV

OPTIMAL NUMBER OF FEATURES AS A FUNCTION OF SAMPLE SIZE IN EXPRESSION-BASED CLASSIFICATION*

This chapter investigates the relationship between the optimal number of features and sample size for expression-based classification. We first give an overview on the history of this problem, then discuss altogether eight classifiers. Analytical approach is applied to quadratic discriminant analysis (QDA), and simulation-based approach is applied to other seven classifiers.

A. Problem Overview

Two-class classification involves a classifier Ψ, a feature vector $\boldsymbol{X} = (X_1, X_2, \ldots, X_d)$ composed of random variables, and a binary random variable Y to be predicted by $\Psi(\boldsymbol{X})$. The values, 0 or 1, of Y are treated as class labels. The error $\varepsilon = P(\Psi(\boldsymbol{X} \neq Y))$ is the probability that the classification is erroneous. The optimal classifier $\Psi_\bullet$ minimizes the probability, $P(\Psi(\boldsymbol{X}) \neq Y)$, of misclassification over all classifiers Ψ. $\Psi_\bullet$ and $\varepsilon_\bullet = P(\Psi_\bullet(\boldsymbol{X}) \neq Y)$ are called the *Bayes classifier* and *Bayes error*, respectively.

Classification via different gene-expression patterns estimated from the microarray images requires designing a classifier (decision function) that takes a vector of gene expression levels as feature vector, and outputs a class label, which predicts the class containing the input vector. It can be between different kinds of cancer, different stages of tumor development, or a host of such differences. The classifier has to be designed based on the available samples.

---

*This chapter contains material reprinted from Pattern Recognition, J. Hua, Z. Xiong, and E.R. Dougherty, "Determination of the optimal number of features for quadratic discriminant analysis via the normal approximation to the discriminant distribution," to appear, Copyright(2004), with permission from Elsevier.

The feature vector consists of the expression levels of a certain number of genes, i.e., features. Given the joint feature-label distribution, increasing the number of features always results in decreased classification error; however, this is not the case when a classifier is designed via a classification rule from sample data. The designed classifier $\Psi_{d,n}$ is now associated with sample size n and feature size d. Typically, for fixed sample size n, the expect error of a designed classifier $\Psi_{d,n}$ decreases and then increases as the number of features d grows. This peaking phenomenon was first rigorously demonstrated for discrete classification [72], but it can affect all classifiers, the manner depending on the feature-label distribution. The problem is especially acute when sample size is very small and the potential number of features is very large. This is precisely the situation in genomic signal processing when one wishes to design gene-expression-based classifiers based on microarray data to discriminate between phenotypes [73, 74]. Given the importance of discovering expression-based genetic markers for disease diagnosis, in particular, cancer [75, 76, 77, 78], in conjunction with the small samples sizes common for microarray studies, determining an appropriate number of features is a critical issue.

The issue is complicated by the fact that if we have D potential features, there are $C(D, d)$ feature sets of size d, and all of these must be considered to assure we have the optimal feature set among them [79]. Owing to the combinatorial intractability of checking all feature sets, numerous algorithms have been developed to find good suboptimal feature sets; nevertheless, when there is a large number of potential features for classification, feature selection is problematic, and the best method depends on the circumstances. Evaluation of methods is generally comparative and based on simulations [80, 81]. To mitigate the confounding effect of feature selection, we will make the simplifying assumption on the covariance matrix of features to circumvent the feature selection procedure. For example, one simplest way is to let any d features

possess the same distribution. This assumption has been stated as optimal feature selection "when all features are equally effective, or when the features are unordered and added in a random way" [82]. To model the situation in which subsets of genes are co-regulated and correlation is internal to these subsets, we can further assume that the covariance matrix of the features is blocked, with each block corresponding to a group of correlated features, and that feature selection follows the order of the features in the covariance matrix (details in later sections). While this does not necessarily produce the optimal feature set for each size, it does provide comparison of the classification rules relative to a global selection procedure that takes into account correlation – as opposed to the less realistic assumption of equal marginal distributions. Under this assumptions on the covariance matrix, the problem can be posed in the following way. Given feature-label distributions $F_1(\boldsymbol{X}^{(1)}, Y), F_2(\boldsymbol{X}^{(2)}, Y), \ldots$, where $\boldsymbol{X}^{(d)} \in \Re^d$, and a classification rule $\boldsymbol{\Psi} = (\Psi_{1,n}, \Psi_{2,n}, \ldots)$, find d to optimize $\Psi_{d,n}$ for a given sample size n. Optimization of $\Psi_{d,n}$ means choosing the number of features so that the expected error of the designed classifier is minimal.

This optimization can be further explained with the notion of *design cost*. If we denote the Bayes error of d features by ε_d, and the error of the designed classifier $\Psi_{d,n}$ by $\varepsilon_{d,n}$, there is a *design cost* $\Delta_{d,n} = \varepsilon_{d,n} - \varepsilon_d$, where $\varepsilon_{d,n}$ and $\Delta_{d,n}$ are sample-dependent random variables. The expected design cost is $E[\Delta_{d,n}]$, the expectation over all possible sample distributions. The expected error of $\Psi_{d,n}$ is decomposed according to

$$E[\varepsilon_{d,n}] = \varepsilon_d + E[\Delta_{d,n}]$$

If the classification rule is consistent, then $E[\Delta_{d,n}] \to 0$ as $n \to \infty$; however, this is of little consequence in settings where n is small and fixed.

With n fixed and small, the number of features becomes critical. As the number

of features increases, the complexity of the classifier, and therefore the amount of data required for precise design increases. Relative to the Bayes classifier determined from a known distribution, the error decreases with increasing d; however, the error of the designed classifier typically decreases and then increases for increasing d, with the optimal number of features being the number that minimizes $E[\varepsilon_{d,n}]$ (equivalently, $E[\Delta_{d,n}]$).

On the surface, it might appear that one could simply try a number of feature sets of varying sizes and then choose the designed classifier having the least error; however, this approach is not satisfactory for small sample sizes. When a classifier is designed under small sample size, one has to estimate its error using the sample data by one of a number of methods, such as cross-validation, but these methods are very inaccurate in the sense that the expected absolute deviation between the estimated error and the true error is often unacceptably high, the situation being worse for complex classification rules and for increasing numbers of features [83]. Indeed, even though error estimation via resubstitution can be substantially low-biased, the increased variance of cross-validation diminishes its feature-ranking ability to the extent that it may perform no better than resubstitution for feature ranking [84]. Owing to this large deviation variation, trying a numerous feature sets and selecting the one with the lowest estimated error presents a multiple-comparison type problem in which it is likely that some feature-set will have an estimated error far below its true error, and therefore appear to provide excellent classification. Since variation is worse for large feature sets, this could create a bias in favor of large feature sets, which goes directly into the teeth of the peaking phenomenon. Thus, we need some general understanding of the kinds of feature-set sizes that provide good performance for a particular classification rule.

In this study, a set of altogether eight classifiers have been investigated. We first

present the analytical results of the quadratic discriminant analysis (QDA) based on a Gaussian approximation to the discriminant distribution in the next section. Then we compared seven classifiers by both simulation on synthetic distribution models and real patient data in the last section.

B. Analytical Results of Quadratic Discriminant Analysis

In this section we are specifically interested in QDA in the case of unequal covariance matrices, in which case the quadratic discriminant does not reduce to a linear discriminant. We are motivated by the recognition that gene-expression data sets for cancer classification often exhibit different variation characteristics for the different cancer phenotypes being discriminated. Moreover, owing to the small sample sizes typically encountered, a simple classifier such as quadratic discrimination is preferable when class separation is not complex in order to mitigate design error [85] and provide class distinctions that possess biological understandable properties.

Two-class QDA concerns finding the optimal classifier Ψ_d to discriminate between two normal class-conditional distributions, $N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, where d is the number of features, $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1 \in \Re^d$ are the mean vectors, and $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$ are the $d \times d$ covariance matrices. $N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ are conditional distributions according to $\boldsymbol{X}|0 \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $\boldsymbol{X}|1 \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$. We will assume equal class probabilities, $P(0) = P(1) = 0.5$, in which case the Bayes classifier is determined according to the discriminant

$$Q_d(\boldsymbol{X}) = (\boldsymbol{X} - \boldsymbol{\mu}_1)' \, \boldsymbol{\Sigma}_1^{-1}(\boldsymbol{X} - \boldsymbol{\mu}_1) - (\boldsymbol{X} - \boldsymbol{\mu_0})'\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{X} - \boldsymbol{\mu}_0) + \log \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|}. \tag{4.1}$$

with $\Psi_d(\boldsymbol{X}) = 1$ if and only if $Q_d(\boldsymbol{X}) \leq 0$.

In applications, the class-conditional distributions are typically unknown and the discriminant must be estimated from sample data. The standard plug-in rule to design (estimate) the optimal classifier from a feature-label sample of size n is to obtain an estimate $Q_{d,n}$ of Q_d by replacing the mean and covariance matrices by their respective sample mean and covariance matrices. The unbiasedness of these estimators assures good estimation for large sample sizes, but not for small sample sizes. The designed classifier $\Psi_{d,n}$ is determined by the estimated discriminant according to $\Psi_{d,n}(\boldsymbol{X}) = 1$ if and only if $Q_{d,n}(\boldsymbol{X}) \leq 0$. Hence, poor estimation of Q_d results in poor classifier design.

In the special case of equal covariance matrices, the discriminant becomes a linear function, thereby characterizing linear discriminant analysis (LDA). This case has received a great deal of attention. Representation of the distribution of $Q_{d,n}$ goes back five decades [86], as does the discovery of an analytic expression for the expected error $E[\varepsilon_{d,n}]$ under the assumption that the sample is evenly split between the two class-conditional distributions, an assumption we make here [87]. Simulation efforts to discover an optimal number of features go back at least three decades [88, 89, 90].

More relevant to our current work are the efforts to use analytic approximations for the expected error to approximate the optimal number of features. In particular, using an asymptotic (in n) error expansion involving the Mahalanobis distance [91], Jain and Waller have investigated the optimal number of features for LDA [92]. They have applied the expansion to very small sample sizes and have obtained results consistent with simulations. Using a truncated error expression, Fukunaga and Hayes have developed a general approximate representation for $E[\Delta_{d,n}]$ and have applied it to the relation between $E[\Delta_{d,n}]$ and the number of features [93]. Their representation is asymptotic in n in the sense that the truncation is made under the assumption that the estimation of the discriminant is very good, which means a large sample

size. Lastly, Raudys and Pikelis have given an analytic expression for the expected error; however, the expression involves an extremely complicated triple integral and therefore they restrict their application to the case of spherical Gaussians [94].

Our goal here is to find an "essentially" analytic method to produce an error curve as a function of the number of features so that the curve can be minimized to determine an optimal number of features. We will do this by using a normal approximation to the distribution of the estimated discriminant $Q_{d,n}$. Since the mean and variance of $Q_{d,n}$ will be exact, these will provide direct insight into the manner in which the covariance matrices affect the optimal number of features. A key point is that the representations of the mean and variance of the estimated discriminant involve only summations of various parameters, which makes their computation very easy. We will derive the mean and variance of $Q_{d,n}$ from its stochastic representation [11] and compare feature-number optimization using the normal approximation to $Q_{d,n}$ with optimization obtained by simulating the true distribution of $Q_{d,n}$. Optimization via the normal approximation to $Q_{d,n}$ provides enormous computational savings in comparison to optimization via simulation of the true distribution. We will see that feature-number optimization via the normal approximation is very accurate when the covariance matrices differ modestly, but that because the distribution of $Q_{d,n}$ varies significantly from normality when the covariance matrices differ greatly, the optimal number of features based on the normal approximation will exceed the actual optimal number when there is large disagreement between the covariance matrices. Nevertheless, this difference turns out not to be important because the true misclassification error using the number of features obtained from the normal approximation and the number obtained from the true distribution differ only slightly even for significantly different covariance matrices (these numbers being obtained via simulation from the true distribution of $Q_{d,n}$).

1. Normal Approximation to the Discriminant Distribution

For equal class probabilities $P(0) = P(1) = 0.5$, the expected error of $\Psi_{d,n}$ can be decomposed into

$$E[\varepsilon_{d,n}] = \frac{1}{2}\left(E[\varepsilon_{d,n}(1|0)] + E[\varepsilon_{d,n}(0|1)]\right). \tag{4.2}$$

where

$$E[\varepsilon_{d,n}(1|0)] = P\{Q_{d,n} \leq 0 | Y = 0\} \tag{4.3}$$

$$E[\varepsilon_{d,n}(0|1)] = P\{Q_{d,n} > 0 | Y = 1\}. \tag{4.4}$$

In [11], the stochastic representations of the conditional distributions $F(Q_{d,n}|Y = 0)$ and $F(Q_{d,n}|Y = 1)$ have been derived. In this section we first briefly review McFarland and Richards' results, then derive our normal approximation.

To make the following presentation clear, we define $Q^0_{d,n}$ and $Q^1_{d,n}$ as two random variables that are distributed as $F(Q_{d,n}|Y = 0)$ and $F(Q_{d,n}|Y = 1)$, respectively. Hence, deriving the stochastic representations of $F(Q_{d,n}|Y = 0)$ and $F(Q_{d,n}|Y = 1)$ is identical to deriving those of $Q^0_{d,n}$ and $Q^1_{d,n}$.

To describe the stochastic representation of $Q^0_{d,n}$, we first define an auxiliary diagonal matrix $\boldsymbol{\Lambda} = diag(\lambda_1, \ldots, \lambda_d)$ and an auxiliary vector $\boldsymbol{\mu} = \{\mu_1, \ldots, \mu_d\}'$ such that

$$\boldsymbol{\Sigma}_1^{-1/2}\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_1^{-1/2} = \boldsymbol{H}\boldsymbol{\Lambda}\boldsymbol{H}' \tag{4.5}$$

$$\boldsymbol{\mu} = \boldsymbol{H}'\boldsymbol{\Sigma}_1^{-1/2}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1), \tag{4.6}$$

where $\boldsymbol{H}$ is a $d \times d$ orthogonal matrix that diagonalizes $\boldsymbol{\Sigma}_1^{-1/2}\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_1^{-1/2}$. Note that

this is equivalent to applying the nonsingular affine transform

$$S(\boldsymbol{X}) = \boldsymbol{H}'\boldsymbol{\Sigma}_1^{-1/2}(\boldsymbol{X} - \boldsymbol{\mu}_1) \tag{4.7}$$

to the data [95], and the original conditional distributions $N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ are transformed into two new normal distributions $N(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ and $N(0, \boldsymbol{I}_d)$. Since one can always transform any two normal distributions into the equivalent form of $N(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ and $N(0, \boldsymbol{I}_d)$, we can focus our study on the classification of the equivalent form without loss of generality. And since the second conditional distribution $N(0, \boldsymbol{I}_d)$ has zero mean and identity covariance matrix, mean and variance of the first conditional distribution, i.e., $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$, can actually be viewed as the measurements of the distance and balance between the two classes, respectively.

We define the mutually independent normal, chi-square and F-distributed random variables involved in the stochastic representation of $Q_{d,n}^0$: $Z_{gj}, g = 1, 2, j = 1, 2, \ldots, d$, are $2d$ i.i.d. standard normal random variables; $T_l, l = 1, 2$, are two i.i.d. chi-square distributions of $n - d$ degrees of freedom; $F_j,\ j = 1, 2, \ldots, d-1$, are $d - 1$ independent F-distributed random variables where each random variable F_j has $(n - j, n - j)$ degrees of freedom. Then for $j = 1, \ldots, d$, we define the following parameters and random variables:

$$\omega_{3j} = n\left(\frac{\lambda_j}{(n+1)(\lambda_j n + 1)}\right)^{1/2} \tag{4.8}$$

$$\gamma_j = \left(\lambda_j + \frac{1}{n}\right)^{-1/2} \mu_j \tag{4.9}$$

$$\nu_1 \sim \frac{(n-1)(n+1)}{nT_1} \tag{4.10}$$

$$\nu_{2j} \sim \frac{(n-1)(\lambda_j + n^{-1})}{T_2}. \tag{4.11}$$

Then $Q^0_{d,n}$ satisfies the stochastic representation [11]

$$\begin{aligned} Q^0_{d,n} &\sim \frac{1}{2}\sum_{j=1}^{d}\left[\nu_{2j}(\omega_{3j}Z_{1j}+(1-\omega^2_{3j})^{1/2}Z_{2j}+\gamma_j)^2-\nu_1 Z^2_{1j}\right] \\ &\quad +\frac{1}{2}\left[\log\left(\frac{T_2}{T_1}\right)-\log|\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0|+\sum_{j=1}^{d-1}\log F_j\right]. \end{aligned} \tag{4.12}$$

McFarland and Richards show that the stochastic representation of $Q^1_{d,n}$ can be formulated similarly; however, here we represent it in a different way that reveals some interesting relationships between $Q^0_{d,n}$ and $Q^1_{d,n}$ and also makes the subsequent description much simpler. We proceed by considering the same classification problem but exchanging the labels of two classes. In the new problem, it is obvious that the only changes to discriminant functions and their corresponding auxiliary variables $\tilde{Q}^0_{d,n}$ and $\tilde{Q}^1_{d,n}$ are the exchanged labels and the flipping signs, i.e., $\tilde{Q}^0_{d,n} \sim -Q^1_{d,n}$ and $\tilde{Q}^1_{d,n} \sim -Q^0_{d,n}$. Moreover, the approach to finding the stochastic representation of the new discriminant $\tilde{Q}^0_{d,n}$ is exactly the same as that for $Q^0_{d,n}$. Thus if we can find the relationship between $\tilde{Q}^0_{d,n}$ and $Q^0_{d,n}$, we can naturally represent $Q^1_{d,n}$ through $Q^0_{d,n}$.

Again we define an auxiliary diagonal matrix $\tilde{\boldsymbol{\Lambda}} = diag(\tilde{\lambda}_1,\ldots,\tilde{\lambda}_d)$ and an auxiliary vector $\tilde{\boldsymbol{\mu}} = \{\tilde{\mu}_1,\ldots,\tilde{\mu}_d\}'$ such that

$$\boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_0^{-1/2} = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{H}}' \tag{4.13}$$

$$\tilde{\boldsymbol{\mu}} = \tilde{\boldsymbol{H}}'\boldsymbol{\Sigma}_0^{-1/2}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0), \tag{4.14}$$

where $\tilde{\boldsymbol{H}}$ is a $d\times d$ orthogonal matrix that diagonalizes $\boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_0^{-1/2}$.

By the derivation listed in Appendix A, we have

$$\tilde{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}^{-1} \tag{4.15}$$

$$\tilde{\boldsymbol{\mu}} = -\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\mu}. \tag{4.16}$$

From Eqs. (4.15) and (4.16) we can see that the stochastic representation of $\tilde{Q}^0_{d,n}$ can be easily obtained from Eq. (4.12) by replacing $\mathbf{\Lambda}$ and $\boldsymbol{\mu}$ with $\mathbf{\Lambda}^{-1}$ and $-\mathbf{\Lambda}^{-1/2}\boldsymbol{\mu}$, respectively. Thus

$$\begin{aligned} Q^1_{d,n}(\boldsymbol{\mu}, \mathbf{\Lambda}) &\sim -\tilde{Q}^0_{d,n}(\boldsymbol{\mu}, \mathbf{\Lambda}) \\ &\sim -Q^0_{d,n}(-\mathbf{\Lambda}^{-1/2}\boldsymbol{\mu}, \mathbf{\Lambda}^{-1}). \end{aligned} \tag{4.17}$$

Since $Q^0_{d,n}$ and $Q^1_{d,n}$ are two distributions of different shapes, we use two normal distributions $N(\mu_{Q^0_{d,n}}, \sigma_{Q^0_{d,n}})$ and $N(\mu_{Q^1_{d,n}}, \sigma_{Q^1_{d,n}})$ to approximate $Q^0_{d,n}$ and $Q^1_{d,n}$, respectively. To do so, we have to find the mean and variance of each distribution.

By replacing Eqs. (4.8)-(4.11) into Eq. (4.12), and applying some lengthy computations sketched in Appendix B, we obtain the mean and variance of $Q^0_{d,n}$:

$$\mu_{Q^0_{d,n}} = \frac{1}{2}\frac{n-1}{n-d-2}\sum_{j=1}^{d}(\lambda_j + \mu_j^2 - 1) - \frac{1}{2}\sum_{j=1}^{d}\log\lambda_j \qquad d < n-2 \tag{4.18}$$

$$\sigma^2_{Q^0_{d,n}} = \frac{(n-1)^2}{(n-d-2)^2(n-d-4)}\left(\sum_{i=1}^{6} v_i - c\right) + \frac{1}{2}\sum_{j=1}^{d}\psi'\left(\frac{n-j}{2}\right) \quad d < n-4, \tag{4.19}$$

where $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$ is the digamma function and

$$
\begin{aligned}
v_1 &= \frac{1}{2}\left(\frac{n}{n+1}\right)^2\left((n-d-2)\sum_{j=1}^{d}\lambda_j^2 + \sum_{i=1}^{d}\sum_{j=1}^{d}\lambda_i\lambda_j\right)\\
v_2 &= \frac{1}{2}\frac{1}{n^2(n+1)^2}\left((n-d-2)\sum_{i=1}^{d}(n+n\lambda_i+1)^2 + \sum_{i=1}^{d}\sum_{j=1}^{d}(n+n\lambda_i+1)(n+n\lambda_j+1)\right)\\
v_3 &= \frac{1}{(n+1)^2}\left((n-d-2)\sum_{j=1}^{d}\lambda_j(n+n\lambda_j+1) + \sum_{i=1}^{d}\sum_{j=1}^{d}\lambda_i(n+n\lambda_j+1)\right)\\
v_4 &= \frac{n}{n+1}\left((n-d-2)\sum_{j=1}^{d}\mu_j^2\lambda_j + \sum_{i=1}^{d}\sum_{j=1}^{d}\mu_i^2\lambda_j\right)\\
v_5 &= \frac{1}{n(n+1)}\left((n-d-2)\sum_{j=1}^{d}\mu_j^2(n+n\lambda_j+1) + \sum_{i=1}^{d}\sum_{j=1}^{d}\mu_i^2(n+n\lambda_j+1)\right)\\
v_6 &= \frac{1}{2}\left(\sum_{j=1}^{d}\mu_j^2\right)^2 - (n-d-4)\sum_{j=1}^{d}\lambda_j + \frac{1}{2}\left(\frac{n+1}{n}\right)^2(n-2)d\\
c &= \frac{n-d-4}{n-1}\sum_{j=1}^{d}\left(\lambda_j + \mu_j^2 + \frac{n+2}{n}\right)
\end{aligned}
$$

Although the mean and variance of $Q_{d,n}^1$ can be derived in a similar way, they can be obtained more conveniently from $Q_{d,n}^0$ based on our new results given by Eq. (4.17):

$$
\begin{aligned}
\mu_{Q_{d,n}^1}(\boldsymbol{\mu},\boldsymbol{\Lambda}) &= -\mu_{Q_{d,n}^0}(-\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\mu},\boldsymbol{\Lambda}^{-1}) && d < n-2\\
&= \frac{1}{2}\frac{n-1}{n-d-2}\sum_{j=1}^{d}\left(1 - \frac{1}{\lambda_j} - \frac{\mu_j^2}{\lambda_j}\right) - \frac{1}{2}\sum_{j=1}^{d}\log\lambda_j && (4.20)\\
\sigma^2_{Q_{d,n}^1}(\boldsymbol{\mu},\boldsymbol{\Lambda}) &= \sigma^2_{Q_{d,n}^0}(-\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\mu},\boldsymbol{\Lambda}^{-1}) && d < n-4\\
&= \frac{(n-1)^2}{(n-d-2)^2(n-d-4)}\left(\sum_{i=1}^{6}v_i(-\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\mu},\boldsymbol{\Lambda}^{-1}) - c(-\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\mu},\boldsymbol{\Lambda}^{-1})\right) && (4.21)
\end{aligned}
$$

With the means and variances obtained, the normal approximations of $Q_{d,n}^0$

and $Q^1_{d,n}$ are naturally obtained, and the corresponding estimation of expected error $E[\varepsilon_{d,n}]$ is:

$$\widehat{E[\varepsilon_{d,n}]} = \frac{1}{2}\left(\tilde{\Phi}\left(\frac{\mu_{Q^0_{d,n}}}{\sigma_{Q^0_{d,n}}}\right) + \tilde{\Phi}\left(-\frac{\mu_{Q^1_{d,n}}}{\sigma_{Q^1_{d,n}}}\right)\right), \tag{4.22}$$

where

$$\tilde{\Phi}(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{1}{2}u^2}du$$

is the upper tail area of the standard normal distribution. This approximation is not used to find a close estimation of $E[\varepsilon_{d,n}]$, and can be quite biased from it; rather, our objective is to find the optimal number of features and avoid possible overfitting. Since Eq. (4.22) depends on sample size n, number of features d, distance $\boldsymbol{\mu}$ and balance $\boldsymbol{\Lambda}$, it is possible to find the relationship between the optimal number of features and sample size in different situations. Before any computer-based study, just from the means and variances of $Q^0_{d,n}$ and $Q^1_{d,n}$ derived above, we can already observe some interesting phenomena.

2. Determination of the Optimal Number of Features

As mentioned in the first section of this chapter, one way to avoid the confounding effect of feature selection is to assume that any d features possess identical distribution. We first examine the simple but commonly studied case in which all features are uncorrelated:

$$\begin{aligned} \boldsymbol{\mu}_0 &= \mu_0\{1,1,\ldots,1\}', & \boldsymbol{\Sigma}_0 &= \sigma_0^2\boldsymbol{I}_d \\ \boldsymbol{\mu}_1 &= \mu_1\{1,1,\ldots,1\}', & \boldsymbol{\Sigma}_1 &= \sigma_1^2\boldsymbol{I}_d \end{aligned} \tag{4.23}$$

Then

$$\begin{aligned} \boldsymbol{\mu} &= \mu\{1, 1, \ldots, 1\}', & \mu &= \frac{\mu_0 - \mu_1}{\sigma_1} \\ \boldsymbol{\Lambda} &= \lambda \boldsymbol{I}_d, & \lambda &= \frac{\sigma_0^2}{\sigma_1^2}. \end{aligned} \tag{4.24}$$

Then with straightforward calculation, $\mu_{Q^0_{d,n}}$, $\mu_{Q^1_{d,n}}$, $\sigma_{Q^0_{d,n}}$ and $\sigma_{Q^1_{d,n}}$ become

$$\mu_{Q^0_{d,n}} = \frac{d}{2}\left(\frac{n-1}{n-d-2}\left(\lambda + \mu^2 - 1\right) - \log\lambda\right) \tag{4.25}$$

$$\mu_{Q^1_{d,n}} = \frac{d}{2}\left(\frac{n-1}{n-d-2}\left(1 - \frac{1}{\lambda} - \frac{\mu^2}{\lambda}\right) - \log\lambda\right) \tag{4.26}$$

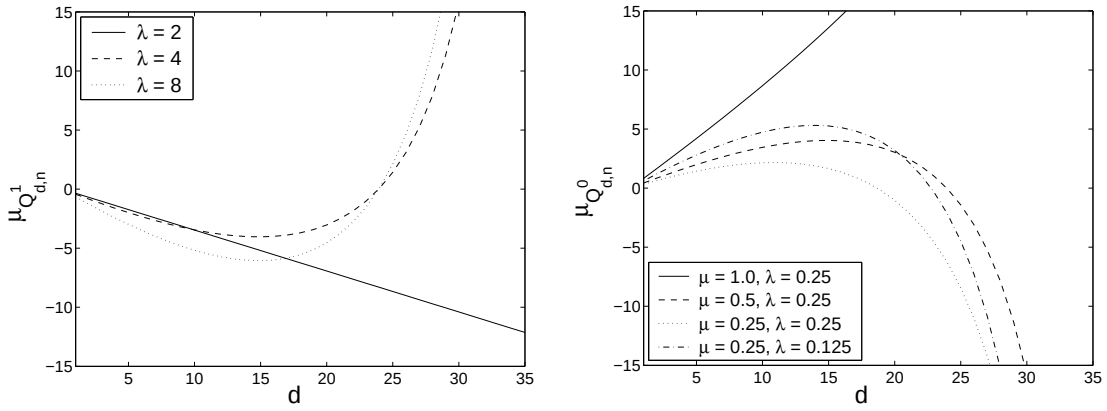
$$\begin{aligned} \sigma^2_{Q^0_{d,n}} =& \frac{d}{2}\frac{(n-1)^2}{n^2(n-d-2)^2(n-d-4)}\left((n-2)\left((1+n\lambda+n\mu^2)^2 + (n+1)^2\right) - 2n^2\mu^4\right) \\ & - \frac{d}{2}\frac{n-1}{n(n-d-2)^2}\left(n(n-1)\mu^4 + 2n^2\lambda + 2n\mu^2 + 2n + 4\right) + \frac{1}{2}\sum_{j=1}^{d}\psi'\left(\frac{n-j}{2}\right) \end{aligned} \tag{4.27}$$

$$\begin{aligned} \sigma^2_{Q^1_{d,n}} =& \frac{d}{2}\frac{(n-1)^2}{n^2(n-d-2)^2(n-d-4)}\frac{1}{\lambda^2}\left((n-2)\left((\lambda+n+n\mu^2)^2 + (n+1)^2\lambda^2\right) - 2n^2\mu^4\right) \\ & - \frac{d}{2}\frac{n-1}{n(n-d-2)^2}\frac{1}{\lambda^2}\left(n(n-1)\mu^4 + 2n^2\lambda + 2n\mu^2\lambda + (2n+4)\lambda^2\right) \\ & + \frac{1}{2}\sum_{j=1}^{d}\psi'\left(\frac{n-j}{2}\right) \end{aligned} \tag{4.28}$$

From Eqs. (4.3) and (4.4) we can infer that to ensure small error rate, $\mu_{Q^0_{d,n}}$ should be much larger than zero, while $\mu_{Q^1_{d,n}}$ much smaller than zero. However, Eqs. (4.25) and (4.26) show that these conditions may not hold when the number of features d increases. When the distributions are very unbalanced, $\mu_{Q^0_{d,n}}$ and $\mu_{Q^1_{d,n}}$ will even flip their signs to the wrong side and induce severe overfitting.

When λ is relatively large, then $1 - \frac{1}{\lambda} - \frac{\mu^2}{\lambda}$ in Eq. (4.26) will be larger than zero. Since $\frac{n-1}{n-d-2}$ increases with d and approaches infinity as d approaches $n-2$, we will

have $\frac{n-1}{n-d-2}\left(1-\frac{1}{\lambda}-\frac{\mu^2}{\lambda}\right)-\log\lambda>0$ when the feature size is large, i.e., $\mu_{Q^1_{d,n}}$ flips its sign to the wrong side and error rate increases dramatically. This phenomenon is shown in Fig. 16 (a), where we fix $n=40$ and $\mu=1$. Three different λ's are considered. For $\lambda=2$, there is $1-\frac{1}{\lambda}-\frac{\mu^2}{\lambda}=0$, and thus $\mu_{Q^1_{d,n}}=-\frac{d}{2}\log 2$ is always less than zero. For $\lambda=4$ or 8, there is $1-\frac{1}{\lambda}-\frac{\mu^2}{\lambda}>0$, and thus $\mu_{Q^1_{d,n}}$ initially decreases, then increases and flips the sign around $d=25$. Comparing to $\mu_{Q^1_{d,n}}$, from Eq. (4.25) we know that $\mu_{Q^0_{d,n}}$ will always be larger than zero and does not flip the sign when λ is relatively large.



(a) (b)

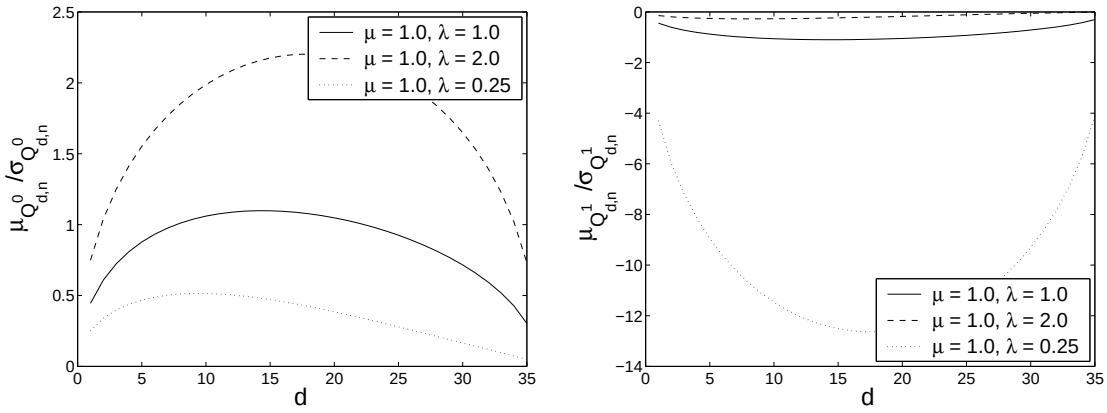
Fig. 16. (a) $\mu_{Q^1_{d,n}}$ vs. d at different λ 's. $n=40$, $\mu=1$; (b) $\mu_{Q^0_{d,n}}$ vs. d at different μ's and λ's. $n=40$.

When λ is relatively small, then $\mu_{Q^1_{d,n}}$ will always be less than zero and does not flip the sign. However, for $\mu_{Q^0_{d,n}}$, from Eq. (4.25) we see that if μ is also relatively small, then $\lambda+\mu^2-1<0$, thus similarly $\mu_{Q^0_{d,n}}$ will flip its sign to the wrong side when feature size is large. This phenomenon is shown in Fig. 16 (b) where $n=40$. Four different pairs of μ and λ are considered. For $\mu=1$, it is obvious that $\lambda+\mu^2-1$ is always larger than zero, thus the sign of $\mu_{Q^0_{d,n}}$ will never flip to the wrong side. As

for other three cases, all of them flip the signs when d is large.

For other cases, $\mu_{Q^0_{d,n}}$ and $\mu_{Q^1_{d,n}}$ will never flip their signs and their absolute values will keep increasing with the feature size. However, Eqs. (4.25)-(4.28) show that overfitting is still hardly avoidable. Take $Q^0_{d,n}$ as a example. When d is small, we have $\frac{1}{n-d-2} \simeq \frac{1}{n-d-4} \simeq \frac{1}{n}$. Then the increasing rate of $\mu_{Q^0_{d,n}}$ according to feature size d is roughly proportional to d. For $\sigma^2_{Q^0_{d,n}}$, the last term $\sum_{j=1}^{d} \psi'(\frac{n-j}{2})$ can be approximated by $\sum_{j=1}^{d} \frac{1}{n-j}$ by using Stirling's approximation and its derivatives. This is a quite small term comparing to other terms and can be omitted for current consideration. Then the increasing rate of $\sigma_{Q^0_{d,n}}$ is proportional to $\sqrt{d}$. Since $\mu_{Q^0_{d,n}}$ increases faster than $\sigma_{Q^0_{d,n}}$ as d increases, the error rate $P(1|0)$ will decrease. When d is large, we have $d \simeq n$. Then the increasing rate of $\mu_{Q^0_{d,n}}$ according to feature size d is roughly proportional to $\frac{1}{n-d-2}$. For $\sigma^2_{Q^0_{d,n}}$, although the first two terms are both at $O(n^3)$, the first term will dominate due to its extra coefficient $\frac{1}{n-d-4}$, and hence the increasing rate of $\sigma_{Q^0_{d,n}}$ is proportional to $\frac{1}{n-d-2}\frac{1}{\sqrt{n-d-4}}$. Since $\mu_{Q^0_{d,n}}$ now increases slower than $\sigma_{Q^0_{d,n}}$, the error rate $P(1|0)$ will increase. This shows that overfitting happens when feature size is large and the error rate $P(1|0)$ must reach its minimum at some feature size between 1 and $n-4$. Fig. 17 shows $\frac{\mu_{Q^0_{d,n}}}{\sigma_{Q^0_{d,n}}}$ and $\frac{\mu_{Q^1_{d,n}}}{\sigma_{Q^1_{d,n}}}$ for three different pairs of μ and λ whose corresponding $\mu_{Q^0_{d,n}}$ and $\mu_{Q^1_{d,n}}$ do not flip signs when feature size increases. It is clear that all of these cases have overfittings when the feature size is large. And comparing to the linear discriminant classifier, which has the optimal feature size at $n-1$ [92], our study shows that the optimal feature size for quadratic discriminant classifier is smaller.

Now we further consider another more general case where the covariance matrices



(a) (b)

Fig. 17. (a) $\frac{\mu_{Q^0_{d,n}}}{\sigma_{Q^0_{d,n}}}$ vs. d at different μ's and λ's. $n = 40$; (b) $\frac{\mu_{Q^1_{d,n}}}{\sigma_{Q^1_{d,n}}}$ vs. d at different μ's and λ's. $n = 40$.

have the same correlation among all features in both classes:

$$\begin{aligned} \boldsymbol{\mu}_1 = \mu_0\{1, 1, \ldots, 1\}', \qquad & \boldsymbol{\Sigma_0} = \sigma_0^2 \begin{bmatrix} 1 & \rho & \ldots & \rho \\ \rho & 1 & \ldots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \ldots & 1 \end{bmatrix} \\ \boldsymbol{\mu}_1 = \mu_1\{1, 1, \ldots, 1\}', \qquad & \boldsymbol{\Sigma_1} = \frac{\sigma_1^2}{\sigma_0^2}\boldsymbol{\Sigma_0} \end{aligned} \tag{4.29}$$

It is obvious that in this case all features are still equivalent and any k features possess the same distribution.

Again, let $\lambda = \frac{\sigma_0^2}{\sigma_1^2}$. Since

$$\begin{aligned} \boldsymbol{\Sigma}_1^{-1/2}\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_1^{-1/2} &= \boldsymbol{\Sigma}_1^{-1/2}(\lambda\boldsymbol{\Sigma}_1)\boldsymbol{\Sigma}_1^{-1/2} \\ &= \lambda\boldsymbol{\Sigma}_1^{-1/2}(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_1^{1/2})\boldsymbol{\Sigma}_1^{-1/2} \\ &= \lambda\boldsymbol{I}_d, \end{aligned} \tag{4.30}$$

by comparing it to Eq. (4.5) we have

$$\boldsymbol{\Lambda} = \lambda \boldsymbol{I}_d \tag{4.31}$$

$$\boldsymbol{H} = \boldsymbol{I}_d. \tag{4.32}$$

Actually, for any pair of $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$ that obeys $\boldsymbol{\Sigma}_0 = \lambda \boldsymbol{\Sigma}_1$, we have $\boldsymbol{\Lambda} = \lambda \boldsymbol{I}_d$.

Since it can be shown that

$$\boldsymbol{\Sigma}_1^{-1/2} = \frac{1}{\sigma_1} \begin{bmatrix} a & b & \dots & b \\ b & a & \dots & b \\ \vdots & \vdots & \ddots & \vdots \\ b & b & \dots & a \end{bmatrix},$$

where

$$\begin{aligned} a &= \frac{1}{d}\left[\frac{d-1}{\sqrt{1-\rho}} + \frac{1}{\sqrt{1+(d-1)\rho}}\right] \\ b &= \frac{1}{d}\left[-\frac{1}{\sqrt{1-\rho}} + \frac{1}{\sqrt{1+(d-1)\rho}}\right], \end{aligned}$$

through Eq. (4.6) we have

$$\begin{aligned} \boldsymbol{\mu} &= \boldsymbol{\Sigma}_1^{-1/2}(\boldsymbol{\mu}_0 - \boldsymbol{\mu}_1) \\ &= \frac{\mu}{\sigma_1\sqrt{1+(d-1)\rho}}\{1, 1, \dots, 1\}', \end{aligned} \tag{4.33}$$

where $\mu = \mu_0 - \mu_1$. Owing to the presence of correlation between features, the distance vector $\boldsymbol{\mu}$ is now a function of feature size d. The larger the correlation, the smaller the distance between classes. And due to the presence of correlation, when feature size increases, the distance at each feature decreases and the total distance becomes

$$|\boldsymbol{\mu}|^2 = \frac{\mu^2}{\sigma_1^2\left(\rho + \frac{1-\rho}{d}\right)}$$

which approaches $\frac{\mu^2}{\sigma_1^2\rho}$ when d is large. This implies that the optimum number of features is even smaller when there is correlation among features.

3. Experimental Results

To verify the accuracy of our proposed normal approximation to determine the optimal feature-set size, we have conducted a series of simulations on various conditions.

Figs. 18-21 show the simulation results obtained on the uncorrelated-feature case defined by Eq. (4.23). Since the classification problem is determined by the two parameters μ and λ, simulations have been conducted with different μ's and λ's. We have varied μ from $\frac{1}{8}$ to 1 and λ from $\frac{1}{8}$ to 8. For each pair of μ and λ, we found the optimal feature sizes at different sample sizes from $n = 10$ up to 100. At each sample size n, our normal approximation calculates $\widehat{E[\varepsilon_{d,n}]}$ at $d = 1, 2, \ldots, n-5$ and finds the optimal number of feature that minimizes $\widehat{E[\varepsilon_{d,n}]}$. To verify the accuracy of the normal approximation, Monte Carlo simulation is conducted to obtain the experimental optimal feature size. For each feature size d, 100000 realizations of $Q_{d,n}^0$ and $Q_{d,n}^1$ are generated separately according to the stochastic representations provided by Eqs. (4.12) and (4.17). Since the independent random variables used to generate $Q_{d,n}^0$ and $Q_{d,n}^1$ increase dramatically with d, for each n, the Monte Carlo simulation is only conducted at a range of feature sizes from 1 to $d = \lfloor \frac{N}{2} \rfloor + 1$. Our simulations show overfittings in all cases, which means that the simulations have covered the ranges where the real optimal feature sizes are located. The optimal feature size is the one giving the smallest misclassification error among the simulation results. When several feature sizes have the same smallest misclassification error, which is possible when the misclassification error is very small, we simply pick the smallest size as the optimal feature size. This may cause some down-biased estimation of the optimal feature size; however, it is of no consequence to us here because we are interested in

finding the number features producing minimal error.

In all figures, the mesh grids are the misclassification errors obtained through the Monte Carlo simulation. The solid lines are those with the lowest error rate and hence the ones showing the optimal feature size based on simulation. The dash lines are the ones based on our normal approximation. In Figs. 18 and 19, we have fixed μ and varied λ and n. In Figs. 20 and 21, we have fixed n and varied μ and λ. We see that the optimal feature size obtained through the normal approximation is very accurate when λ is not too large or too small, i.e., when the covariance matrices differ modestly. But when the covariance matrices of two classes are significantly unbalanced, the distribution of $Q_{d,n}$ varies significantly from normality and $E[\varepsilon_{d,n}]$ is dominated by either $E[\varepsilon_{d,n}(1|0)]$ or $E[\varepsilon_{d,n}(0|1)]$. The optimal number of features based on the normal approximation cannot reflect this and is larger than the simulation-based optimal feature size. However, this difference is not important because the misclassification error using the number of features obtained from the normal approximation and the minimum misclassification error differ only slightly. Specifically, the optimal feature sizes provided by normal simulation are located in the flat regions, for which misclassification errors are small. We have also conducted experiments on the correlated-feature case defined by Eq. (4.29). The results are shown in Figs. 22 and 23. The simulations are conducted analogously to the uncorrelated-feature case, except that there is an identical correlation of $\rho = 0.2$ among all features. From the figures we can see the results are very similar to the uncorrelated-feature case.

A key point is that the normal approximation is easy to implement and obtains the results in almost no time. The Monte Carlo simulation conducted for the uncorrelated-feature case (similarly for the correlated case) runs for about 100 hours and the results are still not smooth when $n \geq 50$, whereas our normal approximation runs in less than 3 seconds on the same computer.

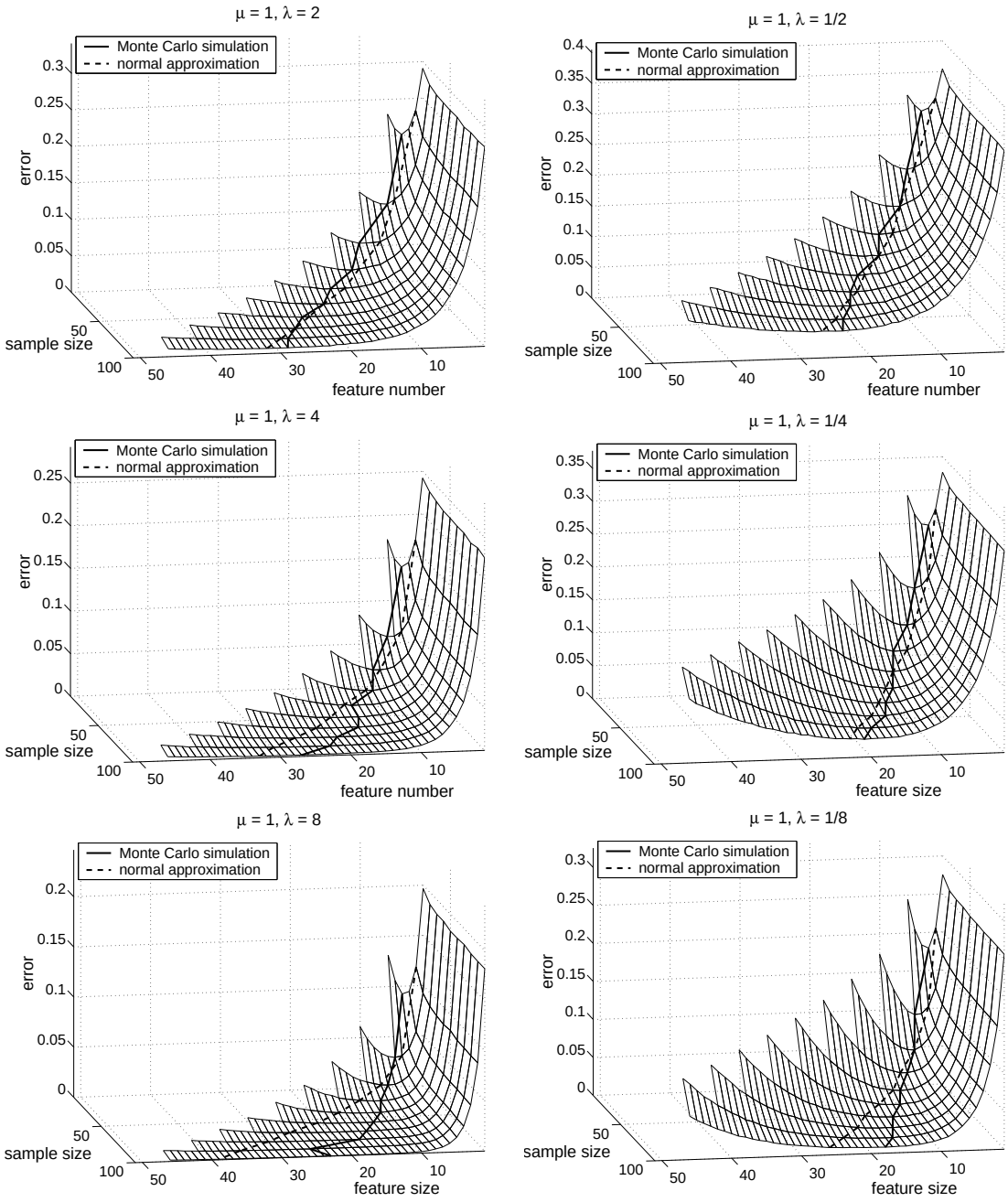


Fig. 18. Optimal feature size at different sample sizes. All features are uncorrelated. $\mu = 1$, and λ varies from $\frac{1}{8}$ to 8.

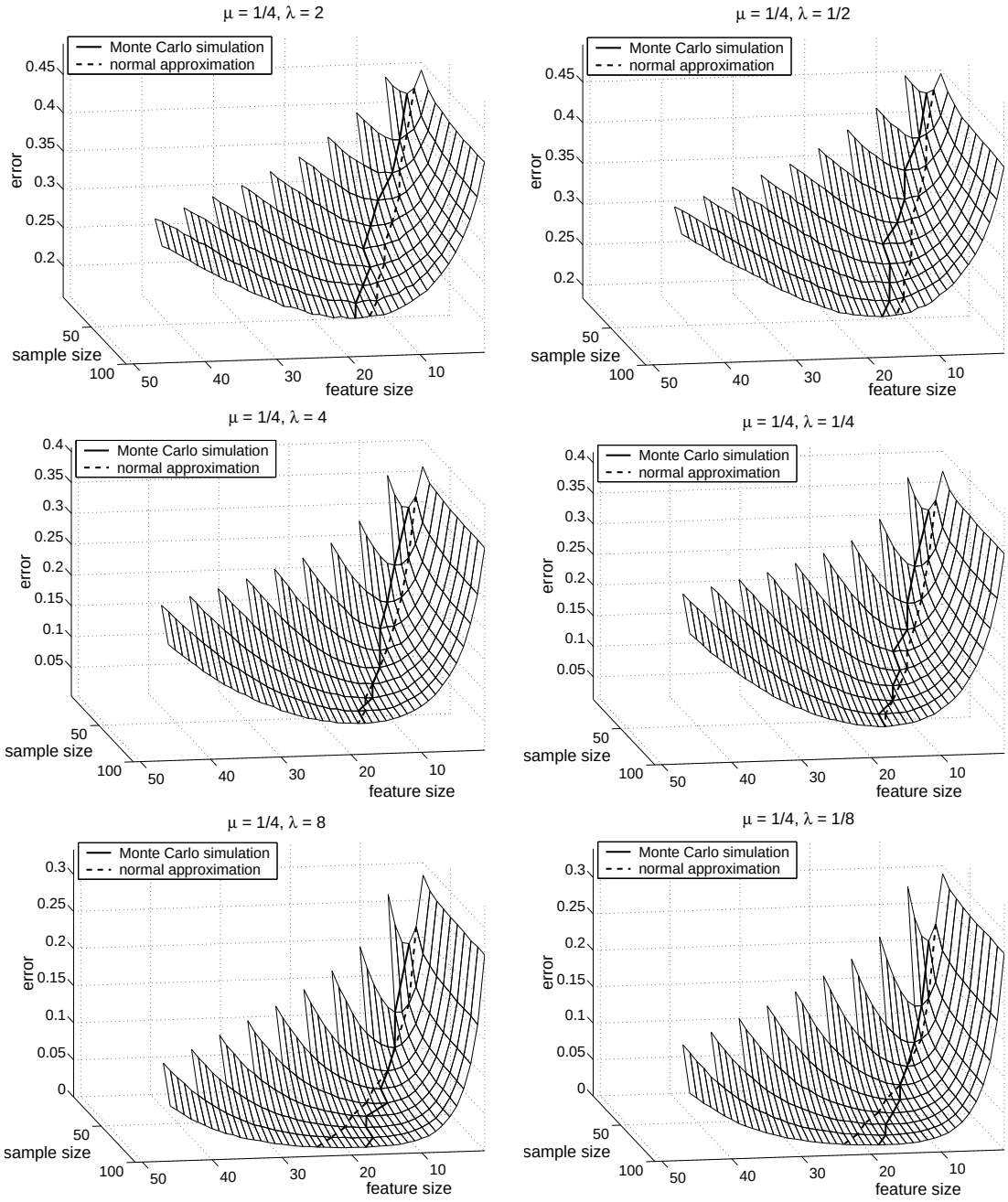


Fig. 19. Optimal feature size at different sample sizes. All features are uncorrelated. $\mu = \frac{1}{4}$, and λ varies from $\frac{1}{8}$ to 8.

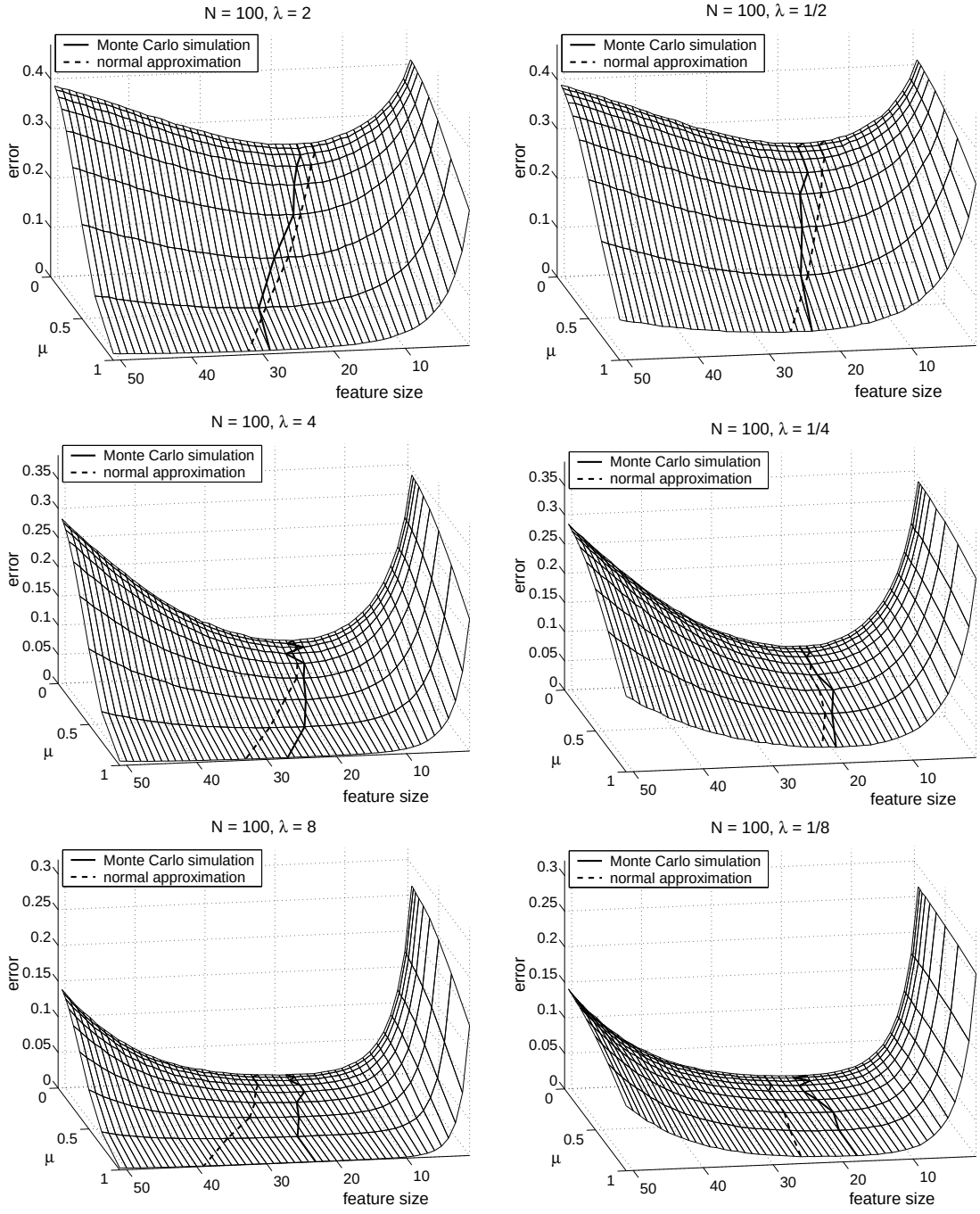


Fig. 20. Optimal feature size at different μ's. All features are uncorrelated. Sample size is fixed at $N = 100$, and λ varies from $\frac{1}{8}$ to 8.

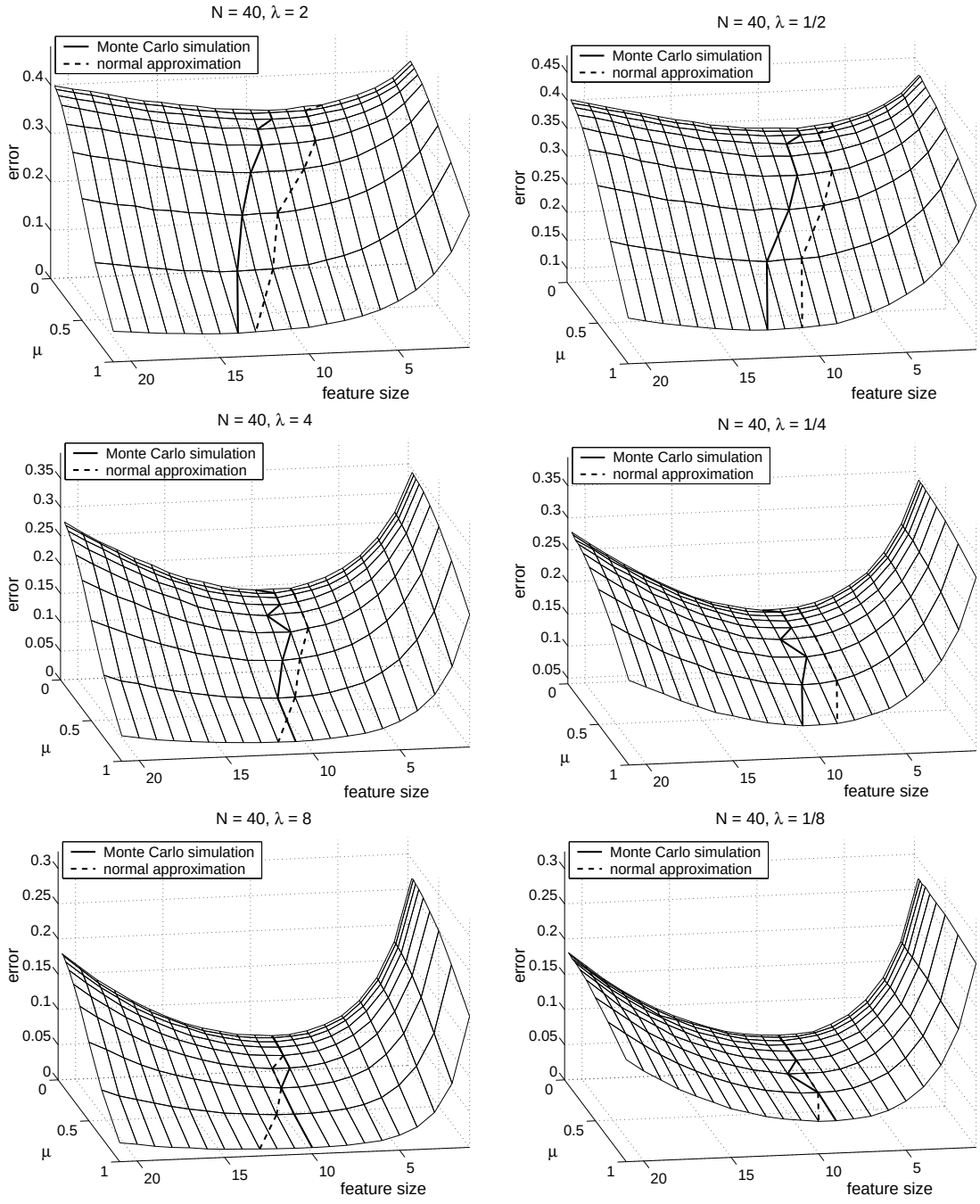


Fig. 21. Optimal feature size at different μ's. All features are uncorrelated. Sample size is fixed at $N = 40$, and λ varies from $\frac{1}{8}$ to 8.

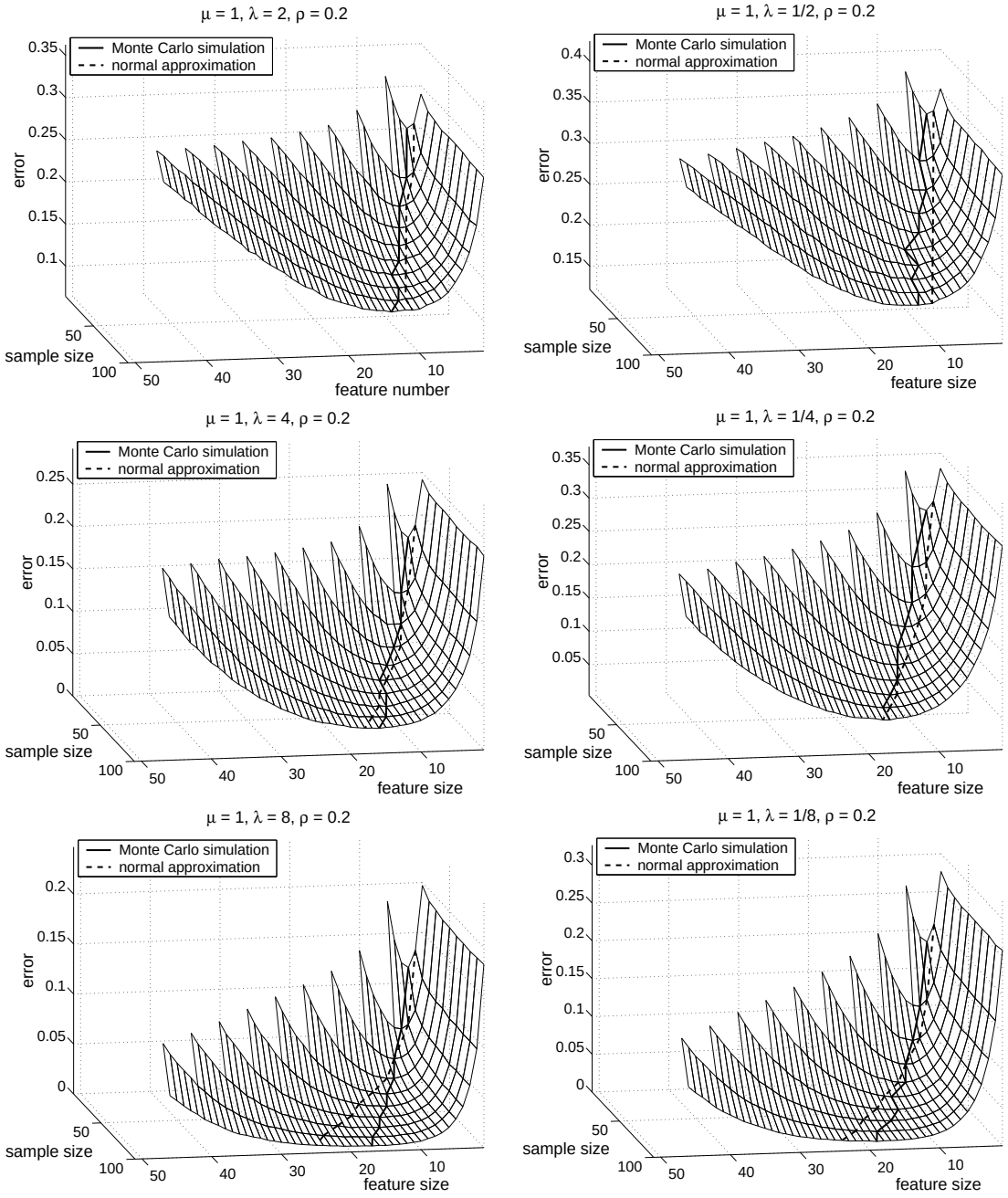


Fig. 22. Optimal feature size at different sample sizes. All features are equally correlated with $\rho = 0.2$. $\mu = 1$, and λ varies from $\frac{1}{8}$ to 8.

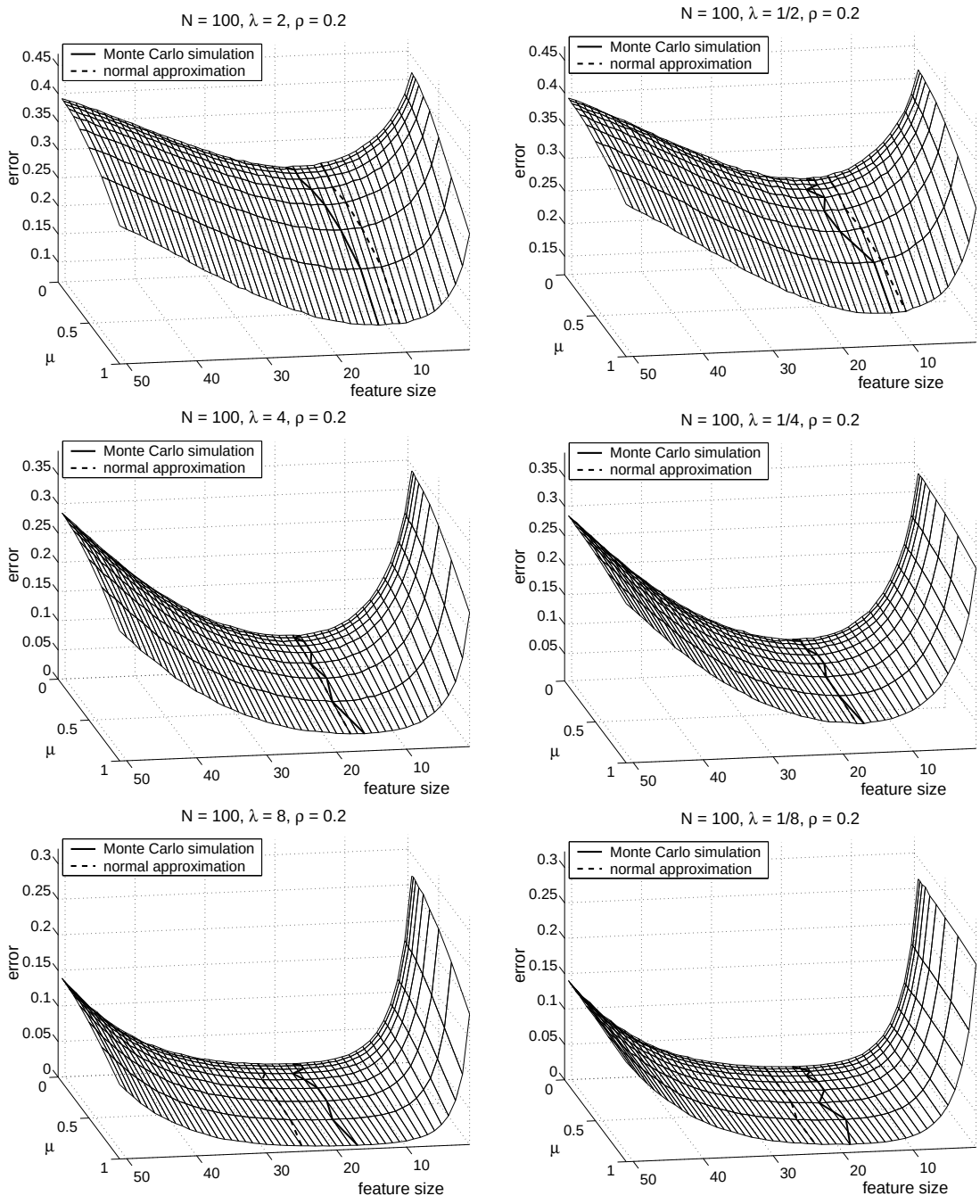


Fig. 23. Optimal feature size at different μ's. All features are equally correlated with $\rho = 0.2$. Sample size is fixed at $N = 100$, and λ varies from $\frac{1}{8}$ to 8.

C. Simulation on Various Classifiers

Although it seems a straightforward approach to find the distribution of the error as a function of the feature-label distribution, number of features, and sample size, only in rare cases this has been achieved. Even for LDA and QDA, the exact forms only exist when the true distributions match the assumptions of the classifiers. This leaves open the simulation route, and this approach has been taken in the past for quadratic and linear discriminant analysis [88, 89, 90]. To do apply simulation for various feature-label distributions and classification rules, and across a wide range of sample and feature-set sizes, requires enormous computation. To achieve the desired end, finding the optimal number of features as a function of sample size, we employ contemporary massively parallel computation. Seven classifiers are considered in our study: 3-nearest-neighbor(3NN), Gaussian kernel, linear support vector machine(Linear SVM), polynomial support vector machine(Polynomial SVM), perceptron, regular histogram and linear discriminant analysis(LDA). For Linear SVM and Polynomial SVM, we use the codes provided by LIBSVM 2.4 [96] with the default setting, except that for Polynomial SVM the degree in the kernel function is set to 6 . For the Gaussian kernel, the smoothing factor h has been set to 0.2 after various trials. For the regular histogram classifier, the cell number along each dimension is set to two or three and evaluated separately, after which the optimal value of the two is selected. Three Gaussian-based models are considered: linear, nonlinear and bimodal, which will be described in detail in the following section. In addition, real patient data from a large breast-cancer study is considered.

Altogether there is a large number of error surfaces for the many cases. These are provided in full on a companion web-site, which is meant to serve as resource for those working with small-sample classification. For the companion web-site, please

visit http://public.tgen.org/tamu/ofs/

1. Simulation Structure for Synthetic Data

We consider three two-class distribution models:

Linear model: The class-conditional distributions are Gaussian, $N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, with identical covariance matrices, $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$. The Bayes classifier is linear and the Bayes decision boundary is a hyperplane. Without loss of generality, we assume that $\boldsymbol{\mu}_0 = (0, 0, \ldots, 0)$ and $\boldsymbol{\mu}_1 = (1, 1, \ldots, 1)$.

Nonlinear model: The class-conditional distributions are Gaussian with covariance matrices differing by a scaling factor, namely, $\lambda\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}$. Throughout the study, $\lambda = 2$. The Bayes classifier is nonlinear and the Bayes decision boundary is quadratic. Again we assume that $\boldsymbol{\mu}_0 = (0, 0, \ldots, 0)$ and $\boldsymbol{\mu}_1 = (1, 1, \ldots, 1)$.

Bimodal model: The class-conditional distribution of class S_0 is Gaussian, centered at $\boldsymbol{\mu}_0 = (0, 0, \ldots, 0)$, and the class-conditional distribution of class S_1 is mixture of two equiprobable Gaussians, centered at $\boldsymbol{\mu}_{10} = (1, 1, \ldots, 1)$ and $\boldsymbol{\mu}_{11} = (-1, -1, \ldots, -1)$. The covariance matrices of the classes are identical. The Bayes decision boundaries are two parallel hyperplanes. Owing to the extreme nonlinear nature of this model, the perceptron, Linear SVM, and LDA classifiers are omitted from our study in this model.

Throughout the study, we assume that the two classes have equal prior probability in all three models. The maximum dimension for all three models is $D = 30$. Hence, the number of features available is less or equal to 30. A consequence of this maximum is that the peaking phenomenon will only show up in the graphs for which peaking is less than or equal to 30 features.

As noted in the section A of this chapter, to avoid the confounding effects of feature selection, we assume a covariance-matrix structure. We let all features have

common variance, like what has been assumed in QDA case, so that the 30 diagonal elements in $\mathbf{\Sigma}$ have the identical value σ^2. Then to set the correlations between features, the 30 features are equally divided into G groups, with each group having $K = 30/G$ features. To divide the features equally, G cannot be arbitrarily chosen. The features from different groups are uncorrelated, and the features from the same group possess the same correlation ρ among each other. If $G = 30$, then all features are uncorrelated. We denote a particular feature with the label $F_{i,j}$, where i, $1 \leq i \leq G$, denotes the group to which the feature belongs and j, $1 \leq j \leq K$, denotes its position in that group. The full feature set takes the form $\boldsymbol{F} = \{F_{1,1}, F_{2,1}, \ldots, F_{G,1}, F_{1,2}, \ldots, F_{G,K}\}$. For any simulation based on a feature subset of d features, the first d features in $\boldsymbol{F}$ are picked. For example, if $G = 10$, then each group has $K = 3$ features, and the covariance matrix, with features ordered as $F_{1,1}, F_{1,2}, F_{1,3}, F_{2,1}, \ldots, F_{10,1}, F_{10,2}, F_{10,3}$, is

$$\mathbf{\Sigma} = \sigma^2 \begin{bmatrix} 1 & \rho & \rho & & & & & & & & \\ \rho & 1 & \rho & & 0 & & \cdot & \cdot & \cdot & & 0 & \\ \rho & \rho & 1 & & & & & & & & & \\ & & & 1 & \rho & \rho & & & & & & \\ & 0 & & \rho & 1 & \rho & \cdot & \cdot & \cdot & & 0 & \\ & & & \rho & \rho & 1 & & & & & & \\ & \cdot & & & \cdot & & \cdot & & & & \cdot & \\ & \cdot & & & \cdot & & & \cdot & & & \cdot & \\ & \cdot & & & \cdot & & & & \cdot & & \cdot & \\ & & & & & & & & & 1 & \rho & \rho \\ & 0 & & & 0 & & \cdot & \cdot & \cdot & \rho & 1 & \rho \\ & & & & & & & & & \rho & \rho & 1 \end{bmatrix}.$$

In this study, all seven classifiers are applied to the three distribution models (except for the perceptron, Linear SVM, and LDA for the bimodal model, as already explained). For each model, altogether 30 different cases are considered according to different covariance-matrix structures and variances:

Variance(σ^2): Three different variances σ^2 are chosen for each model. They correspond to Bayes errors 0.05, 0.10, and 0.15, under the assumption of 10 uncorrelated

features.

Covariance matrix: Four basic covariance-matrix structures are studied by dividing the 30 features into $G = 1$, 5, 10, and 30 groups. For the cases when $G = 1$, 5 and 10, three different correlation coefficients, $\rho = 0.125$, 0.25 and 0.5, are considered. Thus, the total number of covariance-matrix structures studied is 10. Note that for each variance σ^2, different covariance-matrix structures will have different Bayes errors. The increase in correlation among features, either by decreasing G or increasing ρ, will increase the Bayes error for a fixed feature size.

For each case, performances, i.e., error rates, of various classifiers are estimated at various feature sizes and sample sizes based on Monte Carlo simulations:

Feature size (d): Except for the regular histogram, all classifiers are tested on 29 different feature sizes from 2 to 30. For regular histogram, owing to the exponentially increasing cell number, feature sizes are limited from 1 to 10.

Sample size (n): Sample sizes run from 10 to 200, increased by steps of 10, for a total of 20 sample sizes.

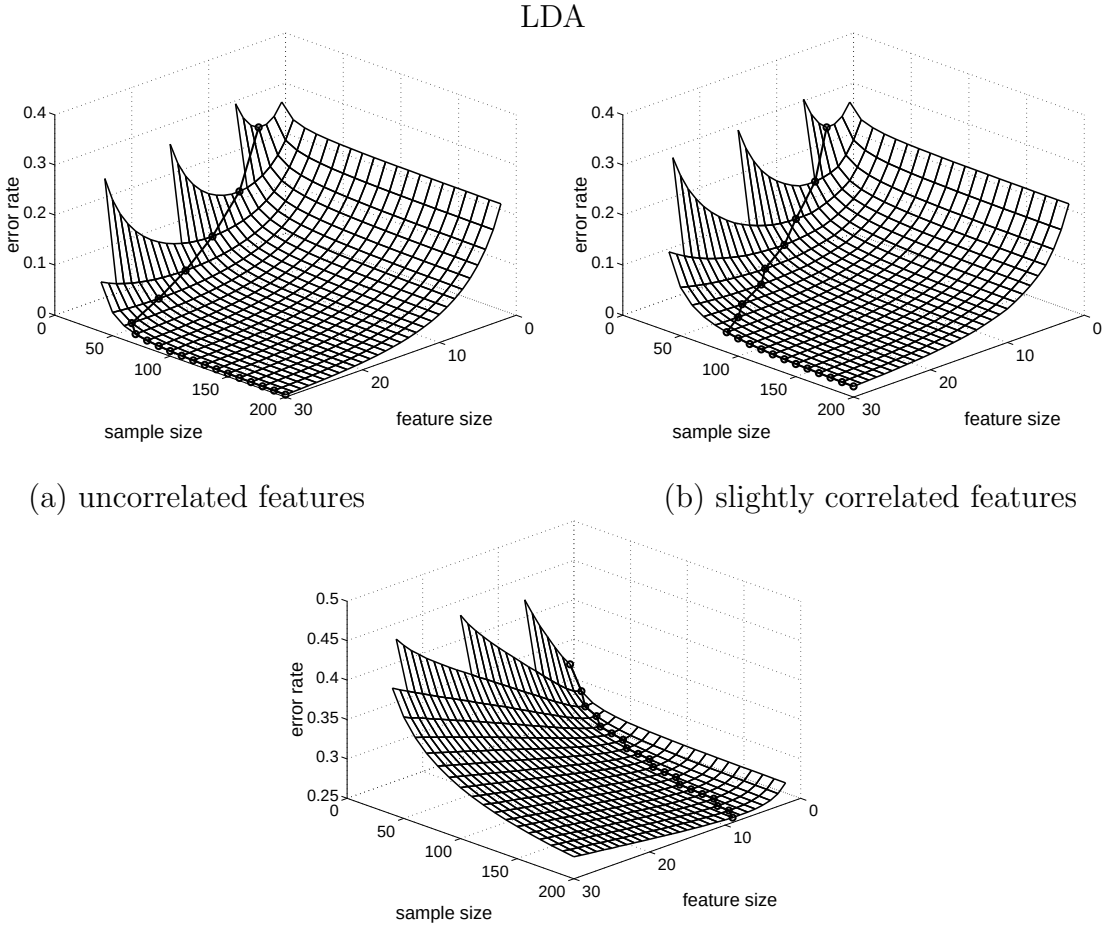
For each feature size d and sample size n, the simulation generates n training samples according to the distribution model, variance, and covariance matrix being tested. The trained classifier is applied to 200 independently generated test samples from the identical distribution. This procedure is repeated $25,000$ times for all classifiers except LDA with $100,000$ repetitions and Linear SVM and Polynomial SVM with $5,000$ repetitions, then the error rates are averaged. The results are presented by a 2-D mesh plot like that in Fig. 24. The black lines with circular markers are those with the lowest error rate, and hence the ones showing the optimal feature size based on the simulation. There is a total of 540 mesh plots on the web-site. In next section we discuss some representative results.

2. Simulation Results on Synthetic Data

Fig. 24 demonstrates the effect of correlation for LDA classification with the linear model. Note that the sample size must exceed the number of features to avoid degeneracy. For uncorrelated features in Fig. 24(a), the optimal feature size is around $n - 1$, which matches well with previously reported LDA results [92]. As the correlation among features increases, the optimal feature size decreases, becoming very small when correlation is high. This also matches results in the same paper, which claims that the optimal feature size is proportional to $\sqrt{n}$ for highly correlated features. In all three parts, uncorrrelated, slightly correlated, and highly correlated, we see the peaking phenomenon and observe the optimal number of features increases with increasing sample size. For microarray-based studies, where sample sizes of less than 50 and feature correlation are commonplace, one should note that with slight correlation, the optimal number of features for $n = 30$ and $n = 50$ is $d = 12$ and $d = 19$, respectively, and with high correlation, the optimal number of features for $n = 30$ and $n = 50$ is $d = 3$ and $d = 4$, respectively. Similar results have been obtained for nonlinear model also.

Fig. 25 provides some results for the regular histogram classifier on the three models. The cell number increases exponentially with feature size and the optimal number of features is quite small in all three models. The curve of the optimal number of features as a function of the sample size shows the common increasing monotonicity. The optimal feature size for the bimodal model is larger, indicating the need for more features to separate three concentrations of mass as opposed to two.

In Figs. 26 and 27, we compare the perceptron, Linear SVM and Polynomial SVM classifiers. Of practical importance are the facts that the Linear SVM shows no peaking phenomenon for up to 30 features, the Polynomial SVM peaks at under



(a) uncorrelated features

(b) slightly correlated features

(c) highly correlated features

Fig. 24. Optimal feature size vs. sample size for LDA classifier. Linear model is tested. σ^2 is set to let Bayes error be 0.05. (a) Uncorrelated features. (b) Slightly correlated features, $G = 5$, $\rho = 0.125$. (c) Highly correlated features, $G = 1$, $\rho = 0.5$.

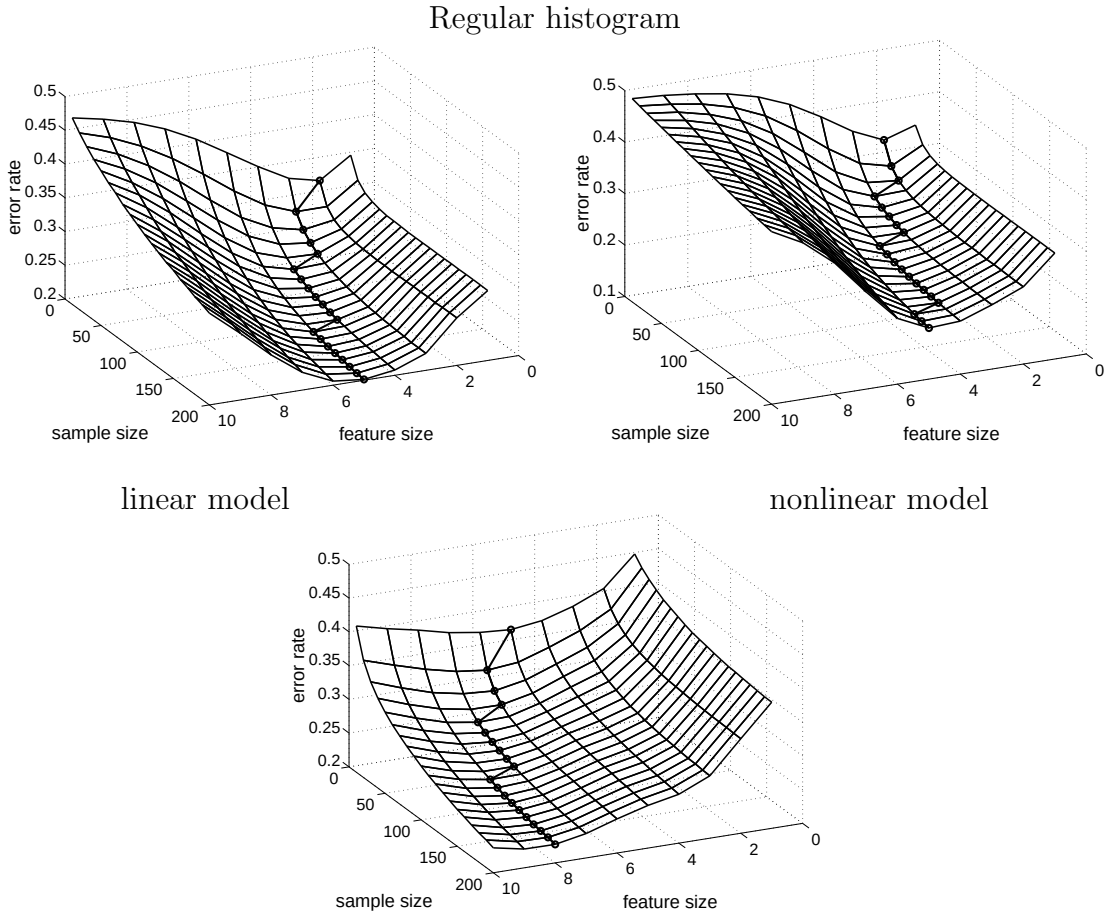


Fig. 25. Optimal feature size vs. sample size for regular histogram classifier. Uncorrelated features. σ^2 is set to let Bayes error be 0.05.

30 only for quite small samples on the uncorrelated linear model and the Polynomial SVM shows no peaking at up to 30 features for the correlated linear model. When there is no peaking, one can safely use a large number of features even for small sample sizes. Note that the optimal-feature-size curves for the perceptron and Linear SVM for the correlated linear and nonlinear models are quite similar, whereas they are very different for the uncorrelated linear model. Note also that the error rate drops much faster relative to sample size for the Polynomial SVM in comparison to the Linear SVM for the correlated model.

Perhaps the most interesting aspect of Figs. 26 and 27 is that there are cases in which the optimal number of features is not monotonically increasing with the sample size (and here we are not referring to slight wobble owing to a flat surface). When it applies, monotonicity follows from the peaking point as the sample size increases. Two phenomena are observed here. For extremely small sample size ($n = 10$), we observe no peaking for the perceptron and Linear SVM except in for the perceptron in the nonlinear model, and the peaking is extremely slight. More striking is that, for the perceptron in all cases and the Linear SVM in the correlated cases, in a range of sample sizes we do not observe the typical concave behavior of the error as a function of the number of features. On the contrary, in some feature size range the classification error will increase and then decrease with the feature size, thereby forming a ridge across the error surface. A zoomed plot for the perceptron in the uncorrelated case in Fig. 28 (a) shows the ridge.

What we are observing can be understood by decomposing the error of the designed classifier into the sum of the error, ε_d, of the optimal classifier for the classification rule relative to the feature-label distribution and the cost, $\Delta_{d,n}$, of designing a classifier from sample S_n: $\varepsilon_{d,n} = \varepsilon_d + \Delta_{d,n}$. Taking expectation with respect to the

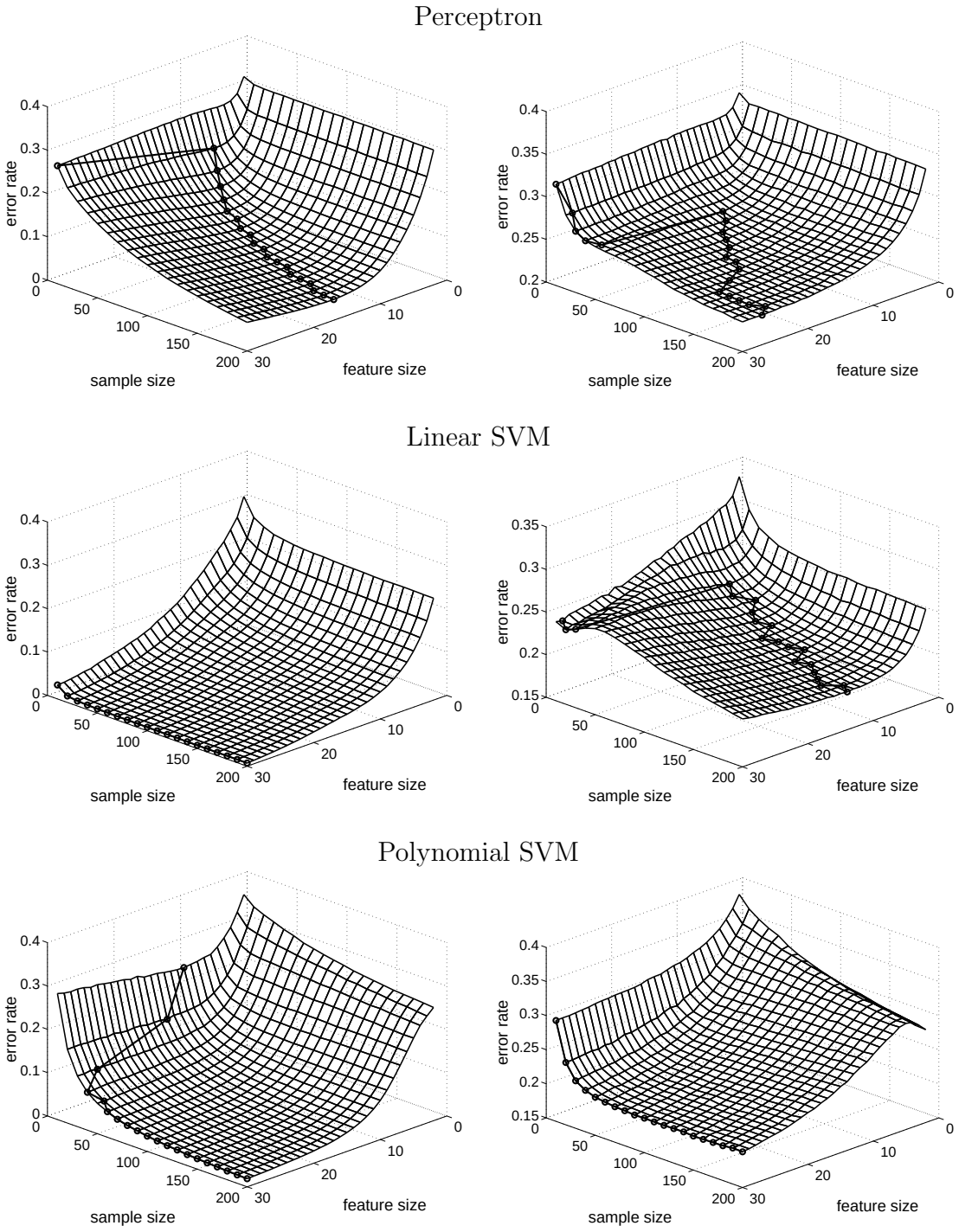


Fig. 26. Optimal feature size vs. sample size for perceptron and SVM classifiers. (a) Linear model, uncorrelated features, σ^2 is set to let Bayes error be 0.05. (b) Linear model, correlated features, $G = 1$, $\rho = 0.25$. σ^2 is set to let Bayes error be 0.05.

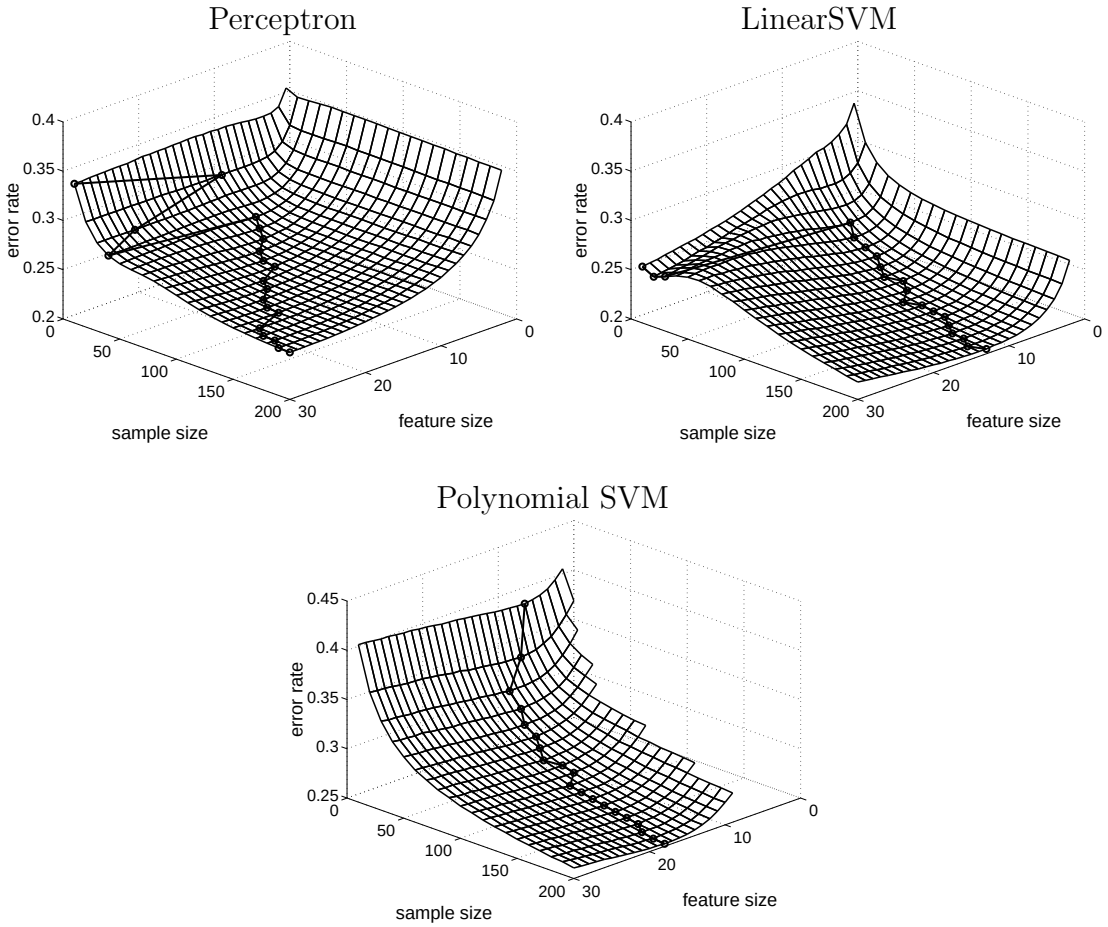


Fig. 27. Optimal feature size vs. sample size for perceptron and SVM classifiers. Nonlinear model, correlated features, $G = 1$, $\rho = 0.25$. σ^2 is set to let Bayes error be 0.05.

distribution of the samples yields

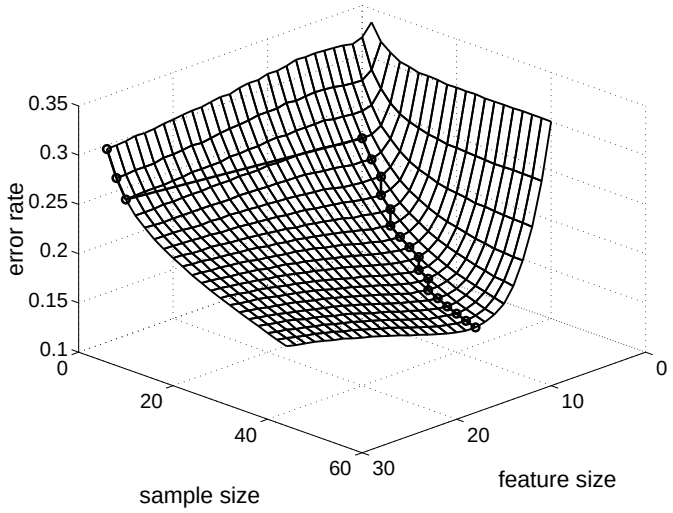
$$E[\varepsilon_{d,n}] = \varepsilon_d + E[\Delta_{d,n}].$$

Considering the expected error as a function of the feature size d, the common interpretation is that $E[\varepsilon_{d,n}]$ decreases to a minimum at d_0 and thereafter increases with increasing d. This means that for $d < d_0$, the optimal error ε_d is falling faster than the design cost $E[\Delta_{d,n}]$ is rising, and that for $d > d_0$, the optimal error ε_d is falling slower than the design cost $E[\Delta_{d,n}]$ is rising. The feature sets for $d < d_0$ are said to *underfit*

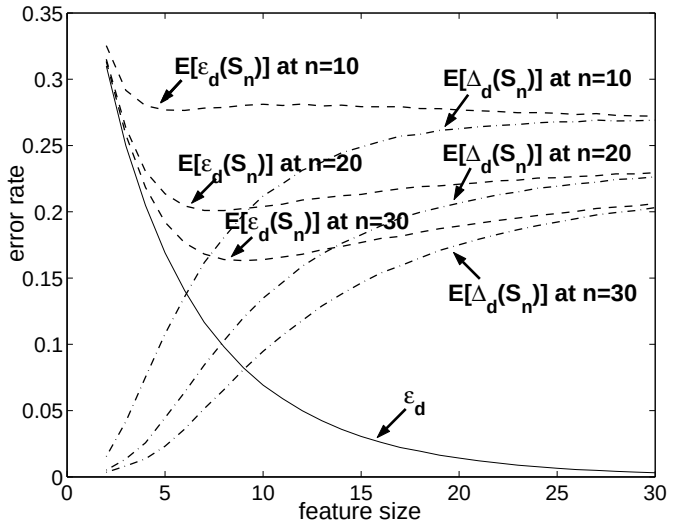
the data because there is insufficient classifier complexity to take full advantage of the data to separate the classes, whereas feature sets for $d > d_0$ are said to *overfit* the data because the complexity of the classifier allows it to produce decision regions that too closely follow the sample points. Under this interpretation, $E[\varepsilon_{d,n}]$ decreasing to a minimum at d_0 and thereafter increasing mean there is decreasing underfitting and then increasing overfitting. The situation may not be so simple. For example, in Fig. 28, we are observing the following phenomenon: there are feature sizes $d_0 < d_1$ such that for $d < d_0$, ε_d is falling faster than $E[\Delta_{d,n}]$ is rising, for $d_0 < d < d_1$, ε_d is falling slower than $E[\Delta_{d,n}]$ is rising, and for $d > d_1$, ε_d is falling faster than $E[\Delta_{d,n}]$ is rising. For sample size $n = 10$, simulation have been run up to 400 features and ε_d is still falling no slower than $E[\Delta_{d,n}]$ is rising. Similar phenomena can be observed for other cases of perceptron and some of the SVM classifiers on the complementary web-site.

In Fig. 29, we compare the 3NN and Gaussian-kernel classifiers on all three distribution models. Since for Gaussian kernel the distance between samples will increase with feature size, the posterior probability of the test sample will be largely determined by the nearest neighbors. Thus, the Gaussian kernel should have similar properties to the 3NN classifier regarding optimal feature size, and this is confirmed by our simulation. A key observation is that for the linear and bimodal models, in which the optimal decision boundaries are flat, there is no peaking up to 30 features. Peaking has been observed at some cases at up to 250 features with sample size $n = 10$, which should have little impact in practical applications.

Perhaps the most interesting observation is that once again we see that the optimal-feature-number curve is not increasing as function of sample size – this being observed in the nonlinear model for both classifiers. The optimal feature size is larger at very small sample sizes, rapidly decreases, and then stabilizes to some constant number as sample increases. To check this stabilization, we have tested the 3NN



(a)



(b)

Fig. 28. A case of perceptron classifier: linear model, uncorrelated features, σ^2 is set to let Bayes error be 0.05. (a) Optimal feature size vs. sample size. (b) Relationship among $E[\varepsilon_d(S_n)]$, $E[\Delta_d(S_n)]$, and ε_d for $n = 10$, 20 and 30.

classifier on the nonlinear model case in Fig. 29 for the sample size up to 5000. The result in Fig. 30 shows that the optimal feature size increases so slowly that it can be practically viewed as a constant unless sample sizes are extremely large. This suggests a useful property of kNN and Gaussian-kernel classifiers: once we find an optimal feature size for a very modest sized sample, we can use the same number of features for much larger samples without sacrificing optimality. Based on our simulations, a corollary of this observation is that using more than a small set of features, say $d \approx 10$, is counterproductive.

3. Real Patient Data

In addition to the synthetic data, we have conducted experimentation based on real patient data. These data come from a microarray-based cancer-classification study [97] that analyzes a large number of microarrays prepared with RNA from breast tumor samples from 295 patients. Using a previously established 70-gene prognosis profile [98], a prognosis signature based on gene-expression is proposed in [97] that correlates well with patient survival data and other existing clinical measures. Of the 295 microarrays, 115 belong to the 'good-prognosis' class, whereas the remaining 180 belong to the 'poor-prognosis' class.

As with the synthetic data, all classifiers are tested on various feature sizes from 1 to 30, , except the regular histogram, which is omitted for the patient data because its error surface is too rough with the limited number of replications used. To mitigate the confounding effects of feature selection, for each feature-set size, floating forward selection [99] is used to find a (hopefully) close-to-optimal feature subset based on all 295 data points. This will provide "population-based" feature sets whose sample-based performances can then be evaluated. To evaluate the performance of each feature subset, we approximate the classification error with a hold-out estimator. For

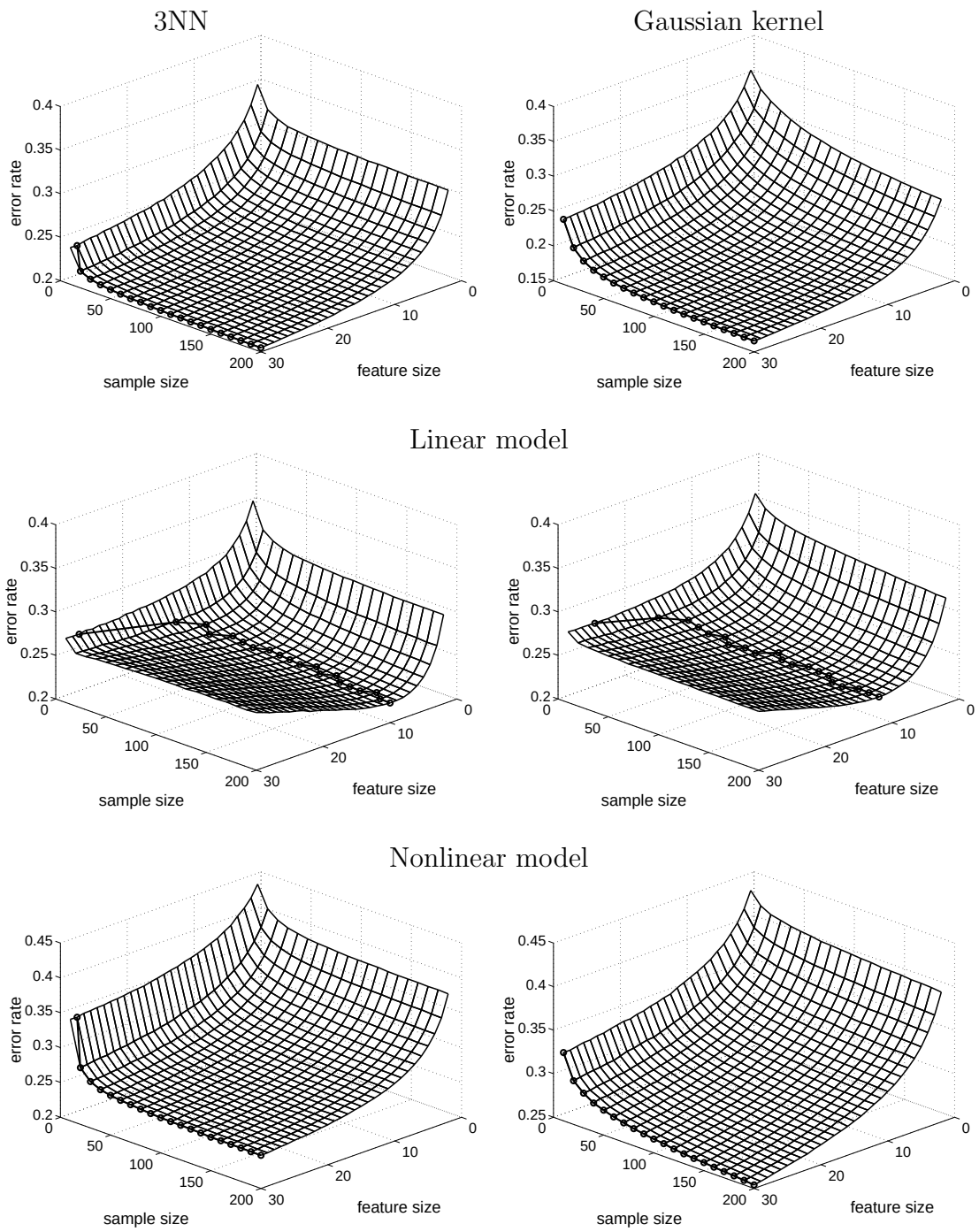


Fig. 29. Optimal feature size vs. sample size for 3NN and Gaussian kernel classifiers. Correlated features, $G = 1$, $\rho = 0.25$. σ^2 is set to let Bayes error be 0.05.

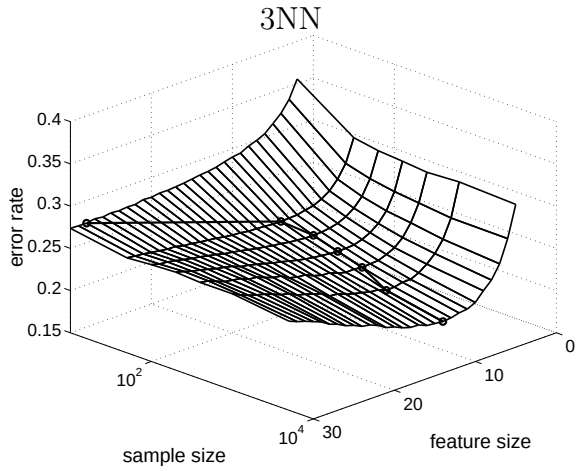


Fig. 30. Optimal feature size vs. sample size for 3NN classifiers. Correlated features, $G = 1$, $\rho = 0.25$. σ^2 is set to let Bayes error be 0.05.

a sample size of n, 1000 sample sets of size n are drawn independently from the 295 data points, and for each observation the different classifiers trained on the n points are tested on the $295 - n$ points not drawn. The 1000 error rates are averaged to obtain an estimate of the sample-based classification error. Since the observations are actually not independent, a large n will induce inaccuracy in the estimation. Hence, we limit n under 40 to reduce the impact of observation correlation. The results are shown in Fig. 31, where all classifiers show some degree of overfitting beginning at feature size from 10 to 20 – some significant and some insignificant. Owing to only 1000 sample sets, there is some wobble in the flat regions of the graphs (especially with the regular histogram), but ignoring this, the results are concordant with the correlated synthetic data. Note especially the flatness of the SVM graphs, especially in the polynomial case, which again indicates the robustness of SVM classification relative to using large feature sets with small samples. Compare this to lack of feature-size robustness for LDA classification. Note once again the similarity of optimal-feature-size performance between the 3NN and Gaussian-kernel classifiers.

Two conclusions can safely be drawn from this study. First, the behavior of the

optimal-feature-size relative to the number of samples depends strongly on the classifier and the feature-label distribution. An immediate corollary is that one should be wary of rules-of-thumb generalized from specific cases. Second, the performance of a designed classifier can be greatly influenced by the number of features and therefore one should attempt to use a number that is in close proximity to the optimal number. This means that it can be useful to refer to a database of optimal-feature-size curves to choose a feature size, even if this means making a necessarily very coarse approximation of the distribution model from the data or making a rough assessment of the correlation. Owing to the roughness of these kinds of approximations, a classifier like the Polynomial SVM, which shows strong robustness with respect to large feature sets, has inherent advantages over a classifier like LDA, which does not show robustness. Our web-site is meant to provide a resource for the community in assessing feature-set sizes.

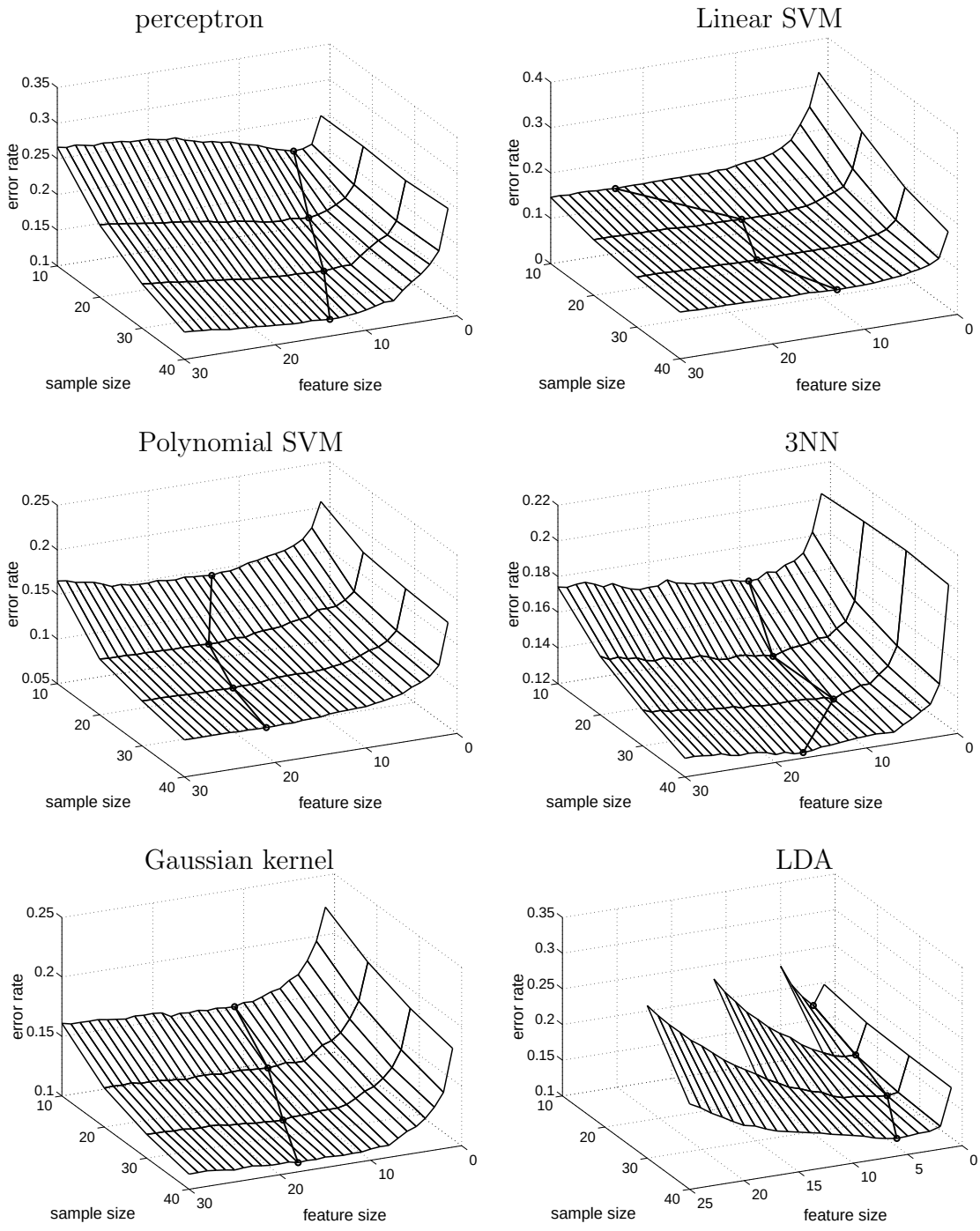


Fig. 31. Error rate vs. feature size for various classifiers on real patient data. Sample size $N = 40$.

CHAPTER V

CONCLUSION

The focus of the this dissertation is genomic image processing for M-FISH and cDNA microarray images. We categorize this research into three topical areas: M-FISH image compression, microarray image processing and expression-based classification.

For M-FISH image compression, we have proposed a new scheme, EMIC, for the highly efficient compression of M-FISH images. EMIC uses shape-adaptive integer wavelet transform and object-based bit-plane coding to generate separate progressive bitstreams for the foreground and background. A specific context model for the arithmetic coding is developed under the design philosophy which can be equally applied to the coding of other types of multi-frame or multispectral images (e.g., MRI and remote-sensing images).

For microarray image processing, we focus on two critical issues: signal estimation and image compression. We have proposed microarray BASICA, which accomplishes segmentation, background adjustment and compression. A fast Mann-Whitney-test-based algorithm with its related post processing procedure are presented for the segmentation, and a novel distortion measure is introduced to help design a new image compression scheme by modifying the EBCOT algorithm.

As for the expression-based classification, we have studied the relationship between optimal number of features and sample size for various classifiers. For QDA, we have developed an essentially analytic method which produces a QDA error curve as a function of the feature size so that the curve can be minimized to determine an optimal number of features. For other classifiers, we have implemented an extensive set of simulations based on both synthetic data that represent some typical cases which might be encountered in the real-life applications, and real patient data.

Our study shows that the behavior of the optimal feature size relative to the number of samples depends strongly on the classifier and the feature-label distribution, and the performance of a designed classifier can be greatly impacted by the number of features. Our web-site is hence meant to provide a resource for the community in assessing feature-set sizes.

Still, our work cannot be viewed as a thorough study of the problem. Contrarily, our research shows that the problem is far more complicated than the common beliefs. Since large sample size is still impossible for most microarray-based genomic studies in the near future due to some practical reasons, it is worthwhile to put more efforts into this problem. Hopefully, some analytical results might be found on certain special cases for more classifiers. Also, the study on the impact of small sample should not be limited to the optimal number of features only. Currently, the impact of small sample on other aspects of expression-based classification, for example, error estimation and feature selection, has already attracted a lot of attention. Furthermore, researchers in other areas of genomic image/signal processing, such as clustering, genetic regulation network, also begin to realize the importance of sample-size related issue. For example, how to evaluate the credibility of the genetic regulation network constructed with the limited samples. All these issues indicate the potential areas of future research.

REFERENCES

[1] A. Borem, F. R. Santos, and D. E. Bowen, *Understanding Biotechnology*, Englewood Cliffs, New Jersey: Prentice Hall PTR, 2003.

[2] N. A. Campbell, J. B. Reece, and L. G. Mitchell, *Biology*, 5th ed., Menlo Park, California: Benjamin Cummings, 1999.

[3] M. R. Speicher, S. G. Ballard, and D. C. Ward, "Karyotyping human chromosomes by combinatorial multi-fluor FISH," *Nature Genetics*, vol. 12, pp. 368-375, April 1996.

[4] E. Schrock, S. du Manoir, T. Veldman, B. Schoell, J. Wienberg, et al., "Multicolor spectral karyotyping of human chromosomes," *Science*, vol.273, pp. 494-497, 1996.

[5] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complimentary DNA microarray," *Science*, vol. 270, pp. 467-470, October 1995.

[6] *The Chipping Forecast*, Supplement to *Nature Genetics*, vol. 21, January 1999.

[7] P. Cosman, R. Gray, and R. Olshen, "Evaluating quality of compressed medical images: SNR, subjective rating and diagnostic accuracy," *Proc. IEEE*, vol. 82, pp. 919-930, June 1994.

[8] J. Ziv and A. Lempel, "Coding theorems for individual sequences via variable-rate coding," *IEEE Trans. Information Theory*, vol. 24, no. 5, pp. 530-536, July 1978.

[9] T. Welch, "A technique for high-performance data compression," *IEEE Computer Magazine*, vol. 17, no. 6, pp. 8-19, June 1984.

[10] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, pp. 1158-1170, July 2000.

[11] H. R. McFarland and D. St. P. Richards, "Exact misclassification probabilities for plug-in normal quadratic discriminant functions II. The heterogeneous case," *Journal of Multivariate Analysis*, vol. 82, pp. 299-330, August 2002.

[12] J. Hua, Z. Xiong, Q. Wu, and K. R. Castleman, "Wavelet-based compression of M-FISH images," *IEEE Trans. on Biomedical Engineering*, to appear.

[13] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard*, New York: Van Nostrand Reihold, 1992.

[14] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*, Boston, Massachusetts: Kluwer Academic Publishers, 2001.

[15] J. Wang and K. Huang, "Medical image compression by using three-dimensional wavelet transformation," *IEEE Trans. Medical Imaging*, vol. 15, pp. 547-554, August 1996.

[16] A. Bilgin, G. Zweig, and M. W. Marcellin, "Three-dimensional image compression using integer wavelet transforms," *Applied Optics: Information Processing*, vol. 39, pp. 1799-1814, April 2000.

[17] Z. Xiong, X. Wu, S. Cheng, and J. Hua, "Lossy-to-lossless compression of medical volumetric data using three-dimensional integer wavelet transforms," *IEEE Trans. Medical Imaging*, vol. 22, pp. 459-470, March 2003.

[18] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. G. I. Nieto, and J. Cornelis, "Wavelet coding of volumetric medical datasets," *IEEE Trans. Medical Imaging*, vol. 22, pp. 441-458, March 2003.

[19] G. Menegaz and J.-P. Thiran, "3D encoding/2D decoding of medical data," *IEEE Trans. Medical Imaging*, vol. 22, pp. 424-440, March 2003.

[20] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445-3463, December 1993.

[21] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 6, pp. 243-250, June 1996.

[22] A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy compression," *IEEE Trans. Image Processing*, vol. 5, pp. 1303-1310, September 1996.

[23] M. Pratt, C. Chu, and S. Wong, "Volume compression of MRI data using zerotrees of wavelet coefficients," in *Proc. of SPIE Wavelet Applications in Signal and Image Processing IV*, Denver, Colorado, USA, vol. 2825, pp. 752-763, September 1996.

[24] J. Luo and C. W. Chen, "Coherently three-dimensional wavelet-based approach to volumetric image compression," *J. Electronic Imaging*, vol. 7, pp. 474-485, July 1998.

[25] S. Wong, L. Zaremba, D. Gooden, and H. Huang, "Radiologic image compression - a review," *Proc. IEEE*, vol. 83, pp. 194-219, February 1995.

[26] J. Hu, Y. Wang, and P. Cahill, "Multispectral code excited linear prediction coding and its application in magnetic resonance images," *IEEE Trans. Image Processing*, vol. 6, pp. 1555-1566, November 1997.

[27] B.-J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate embedded video coding with 3D set partitioning in hierarchical trees (3D SPIHT)," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 10, pp. 1365-1374, December 2000.

[28] J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang, "3-D embedded subband coding with optimal truncation (3-D ESCOT)," *Applied and Computational Harmonic Analysis*, vol. 10, pp. 290-315, May 2001.

[29] V. Vlahakis and R. Kitney, "ROI approach to wavelet-based, hybrid compression of MR images," in *Proc. 6th Intl. Conf. on Image Processing and Its Applications*, Dublin, Ireland, vol. 2, pp. 833-837, July 1997.

[30] Z. Liu, Z. Xiong, Q. Wu, Y. Wang, and K. Castleman, "Cascaded differential and wavelet compression of chromosome images," *IEEE Trans. on Biomedical Engineering*, vol. 49, pp. 372-383, April 2002.

[31] K. Castleman, *Digital Image Processing*, Upper Saddle River, New Jersey: Prentice Hall, 1996.

[32] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, pp. 247-269, July 1998.

[33] R. C. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *J. Applied and Computational Harmonics Analysis*, vol. 5, pp. 332-369, July 1998.

[34] J. Villasenor, B. Belzer, and J. Liao, "Wavelet filter evaluation for image compression," *IEEE Trans. Image Processing*, vol. 4, pp. 1053-1060, August 1995.

[35] M. Adam and F. Kossentni, "Reversible integer-to-integer wavelet transforms for image compression: performance evaluation and analysis," *IEEE Trans. Image Processing*, vol. 9, pp. 1010-1024, June 2000.

[36] M. Antonini, M. Barlaud, P. Mathien, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 205-220, April 1992.

[37] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Lossless image compression using integer to integer wavelet transforms," in *Proc. Intl. Conf. on Image Processing*, Washington DC, USA, vol. 1, pp. 596-599, October 1997.

[38] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Trans. Circuits and Systems for Video Tech*, vol. 10, pp. 725-743, August 2000.

[39] I. Witten, R. Neal, and J. Cleary, "Arithmetic coding for data compression," *Communications of the ACM*, vol. 30, pp. 520-540, June 1987.

[40] J. Hua, Z. Xiong, and X. Wu, "High-performance 3-D embedded wavelet video (EWV) coding," in *Proc. Multimedia Signal Processing Workshop*, Cannes, France, vol. 1, pp. 569-574, October 2001.

[41] T. Cover and J. Thomas, *Elements of Information Theory*, New York: Wiley, 1991.

[42] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Trans. Information Theory*, vol. 30, pp. 629-636, July 1984.

[43] X. Wu, P. A. Chou, and X. Xue, "Minimum conditional entropy context quantization," in *Proc. of Intl. Symp. Inform. Theory*, Sorrento, Italy, vol.1, pp. 43, June 2000.

[44] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, New York: Academic Press, 1999.

[45] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, Boston, Massachusetts: Kluwer Academic Publishers, 1992.

[46] X. Wu, "High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression," in *Proc. of 31st Asilomar Conf. on Signals, Systems, and Computers*, Pacific Grove, California, vol. 2, pp. 1378-1384, November 1997.

[47] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Processing*, Vol.3, No. 5, pp. 572-588, September, 1994.

[48] J. Hua, Z. Liu, Z. Xiong, Q. Wu, and K. R. Castleman, "Microarray BASICA: Background adjustment, segmentation, image compression and analysis of microarray images," *EURASIP Journal on Applied Signal Processing: Special Issue on Genomic Signal Processing*, vol. 4, pp. 92-107, January 2004.

[49] J. Hua, Z. Xiong, Q. Wu, and K. Castleman, "Fast segmentation and lossy-to-lossless compression of DNA microarray images," in *Proc. Workshop on Genomic Signal Processing and Statistics (GENSIPS)*, Raleigh, North Carolina, vol. 1, CP2-01, October 2002.

[50] J. Hua, Z. Xiong, Q. Wu, and K. Castleman, "Microarray BASICA: Background adjustment, segmentation, image compression and analysis of microarray im-

ages," in *Proc. Intl. Conf. on Image Processing*, Barcelona, Spain, vol.1, pp. 585-588, September 2003.

[51] Y. Chen, E. R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *J. Biomedical Optics*, vol. 2, pp. 364-374, October 1997.

[52] Y. Chen, V. Kamat, E. R. Dougherty, M. L. Bitter, P. S. Meltzer, and J. M. Trent, "Ratio statistics of gene expression levels and applications to microarray data analysis," *Bioinformatics*, vol. 18, pp. 1207-1215, August 2002.

[53] Scanalytics, Inc., *MicroArray Suite For Macintosh Version 2.1 Users Guide*, Fairfax, Virginia, USA, August 2001.

[54] Y. H. Yang, M. J. Buckley, S. Dudoit, and T. P. Speed, "Comparison of methods for image analysis on cDNA microarray data," *J. Computational and Graphical Statistics*, vol. 11, pp. 108-136, March 2002.

[55] Raytest Isotopenmessgeraete GmbH, *AIDA Array Metrix User's Manual*, Straubenhardt, Germany, 2002.

[56] Imaging Research Inc., *ArrayVision Version 7.0 Reference Manual*, St. Catharines, Ontario, Canada, 2002.

[57] J. Buhler, T. Ideker, and D. Haynor, "Dapple: Improved techniques for finding spots on DNA microarrays," *UW CSE Technical Report UWTR 2000-08-05*, 2000.

[58] A. J. Carlisle, V. V. Prabhu, A. Elkahloun, J. Hudson, J. M. Trent, et al. "Development of a prostate cDNA microarray and statistical gene expression analysis package," *Molecular Carcinogenesis*, vol. 28, pp. 12-22, January 2000.

[59] Axon Instruments, Inc., *GenePix Pro 4.1 User's Guide and Toturial, Rev G*, Union City, Carlifornia, USA, 2002.

[60] CLONDIAG chip technologies GmbH, *IconoClust 2.1 Manual*, Jena, Germany, 2002.

[61] X. Wang, S. Ghosh, and S. Guo, "Quantitative quality control in microarray image processing and data acquisition," *Nucleic Acids Research*, vol. 29, pp. 75-82, January 2001.

[62] Nonlinear Dynamics Ltd., *Phoretix Array version 3.0 User's Guide*, Newcastle upon Tyne, United Kingdom, 2002.

[63] PerkinElmer Life Sciences, *QuantArray Analysis Software, Operator's Manual*, Boston, Massachusetts, USA, 1999.

[64] M. Eisen, *ScanAlyze User Manual*, Berkeley, California, USA, 1999.

[65] A. N. Jain, T. A. Tokuyasu, A. M. Snijders, R. Segraves, D. G. Albertson, and D. Pinkel, "Fully automatic quantification of microarray image data," *Genome Research*, vol. 12, pp.325-332, February 2002.

[66] R. Jornsten, W. Wang, B. Yu, and K. Ramchandran, "Microarray image compression: SLOCO and the effect of information loss," *Signal Processing: Special Issue on Genomic Signal Processing*, vol. 83, pp. 859-869, April 2003.

[67] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 641-647, June 1994.

[68] E. R. Dougherty, *An Introduction to Morphological Image Processing*, Bellingham, Washington: SPIE Optical Engineering Press, 1992.

[69] B. P. Durbin, J. S. Hardin, D. M. Hawkins, and D. M. Rocke, "A variance-stabilizing transformation for gene-expression microarray data." *Bioinformatics*, vol 18: Suppl. 1, pp. S105-S110, July 2002.

[70] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *IEEE Trans. Image Processing*, vol. 9, pp. 1309-1324, August 2000.

[71] J. Hua, Z. Xiong, and E. Dougherty, "Determination of the optimal number of features for quadratic discriminant analysis via the normal approximation to the discriminant distribution," *Pattern Recognition*, to appear.

[72] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. on Information Theory*, vol. 14, pp. 55-63, January 1968.

[73] E. R. Dougherty, "Small sample issues for microarray-based classification," *Comparative and Functional Genomics*, vol. 2, pp. 28-34, January 2001.

[74] E. R. Dougherty, I. Shmulevich, and M. L. Bittner, "Genomic signal processing: the salient issues," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 146-153, January 2004.

[75] T. R. Golub, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531-537, April 1999.

[76] M. Bittner, et al., "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, pp. 536-540, August 2000.

[77] J. Khan, et al., "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine*, vol. 7, pp.

673-679, June 2000.

[78] I. Hedenfalk, et al., "Gene expression profiles in hereditary breast cancer," *New England Journal of Medicine*, vol. 344, pp. 539-548, February 2001.

[79] T. M. Cover and J. M. Van Campenhout, "On the possible orderings in the measurement selection problem," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 7, pp. 657-661, September 1977.

[80] A. K. Jain and D. Zongker, "Feature selection - evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 153-158, February 1997.

[81] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognition*, vol. 33 pp. 25-41, January 2000.

[82] S. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: recommendations for practitioners," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp.252-264, March 1991.

[83] U. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?," *Bioinformatics*, vol. 20, pp. 374-380, February 2004.

[84] U. Braga-Neto, et al. "Is cross-validation better than resubstitution for ranking genes?," *Bioinformatics*, vol. 20, pp. 253-258, January 2004.

[85] L. Devroye, L Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, New York: Springer-Verlag, 1996.

[86] A. H. Bowker, "A representation of Hotelling's T^2 and Anderson's classification statistics," in *Studies in Item Analysis and Prediction(Edited by H. Solomon)*, pp. 285-292, Palo Alto, Carlifornia: Stanford University Press, 1961.

[87] R. Sitgreaves, "Some results on the distribution of the W-classification," in *Studies in Item Analysis and Prediction(Edited by H. Solomon)*, pp. 241-251, Palo Alto, Carlifornia: Stanford University Press, 1961.

[88] J. W. Van Ness and C. Simpson, "On the effects of dimension in discriminant analysis," *Technometrics*, vol. 18, pp. 175-187, May 1976.

[89] T. S. El-Sheikh and A. G. Wacker, "Effect of dimensionality and estimation on the performance of gaussian classifiers," *Pattern Recognition*, vol. 12, pp. 115-126, January 1980.

[90] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," in *Handbook of Statistics, Vol. II(edited by P. R. Krishnaiah and L.N. Kanal)*, pp. 835-855, Amsterdam: North-Holland, 1982.

[91] R. Sitgreaves, "Some operating characteristics of linear discriminant functions," in *Discriminant Analysis and Applications(Edited by T. Cacoullos)*, pp. 365-374, New York: Academic Press, 1973.

[92] A. K. Jain and W. G. Waller, "On the optimal number of features in the classification of multivariate gaussian data," *Pattern Recognition*, vol. 10, pp. 365-374, May 1978.

[93] K. Fukunaga and R. R. Hayes, "Effects of sample size in classifier design," *IEEE Trans. Pattern Anal. Machine Intelli.*, vol. 11, pp. 873-885, August 1989.

[94] S. Raudys and V. Pikelis, "On dimensionality, sample size, classification error, and complexity of classification algorithm in pattern recognition," *IEEE Trans. Pattern Anal. Machine Intelli.*, vol. 2, pp. 242-252, May 1980.

[95] T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*, 2nd ed., New York: Wiley, 1984.

[96] C.-C. Chang and C.-J. Lin, "LIBSVM: Introduction and benchmarks," *Technical Report, Department of Computer Science and Information Engineering, National Taiwan University*, Taipei, Taiwan, 2000.

[97] M. J. van de Vijver, et al., "A gene-expression signature as a predictor of survival in breast cancer." *New Eng. J. Med.*, vol. 347, pp. 1999-2009, December 2002.

[98] L. J. vant Veer, et al., "Gene expression profiling predicts clinical outcome of breast cancer." *Nature*, vol. 415, pp. 530-536, January 2002.

[99] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, pp. 1119-1125, November 1994.

APPENDIX A

DERIVE THE RELATIONSHIP BETWEEN $Q^1_{D,N}$ AND $Q^0_{D,N}$

To relate $Q^1_{d,n}$ with $Q^0_{d,n}$, our objective is to represent $\tilde{\boldsymbol{\Lambda}}$ and $\tilde{\boldsymbol{\mu}}$ with $\boldsymbol{\Lambda}$ and $\boldsymbol{\mu}$. From Eq. (4.5) we have

$$\begin{aligned}
& (\boldsymbol{\Sigma}_1^{-1/2}\boldsymbol{\Sigma}_0\boldsymbol{\Sigma}_1^{-1/2})^{-1} = (\boldsymbol{H}\boldsymbol{\Lambda}\boldsymbol{H}')^{-1} \\
\Longrightarrow \quad & \boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\Sigma}_1^{1/2} = \boldsymbol{H}\boldsymbol{\Lambda}^{-1}\boldsymbol{H}' \\
\Longrightarrow \quad & \boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_1^{1/2} = \boldsymbol{H}\boldsymbol{\Lambda}^{-1}\boldsymbol{H}' \\
\Longrightarrow \quad & (\boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_1^{-1/2})(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_1^{1/2})(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2}) = (\boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_1^{-1/2})\boldsymbol{H}\boldsymbol{\Lambda}^{-1}\boldsymbol{H}'(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2}) \\
\Longrightarrow \quad & \boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_0^{-1/2} = (\boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_1^{-1/2})\boldsymbol{H}\boldsymbol{\Lambda}^{-1}\boldsymbol{H}'(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2})
\end{aligned} \tag{A.1}$$

Since $\boldsymbol{\Sigma}_0^{-1/2}$ and $\boldsymbol{\Sigma}_1^{1/2}$ are symmetric matrices, $\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2}$ is orthogonal:

$$\begin{aligned}
& (\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2})(\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2})' \\
= \quad & \boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_1^{1/2})\boldsymbol{H}\boldsymbol{\Lambda}^{1/2} \\
= \quad & \boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'(\boldsymbol{H}\boldsymbol{\Lambda}^{-1}\boldsymbol{H}')\boldsymbol{H}\boldsymbol{\Lambda}^{1/2} \\
= \quad & 1.
\end{aligned}$$

Then the right hand side of Eq. (A.1) can be further written as

$$\begin{aligned}
& (\boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_1^{-1/2})\boldsymbol{H}\boldsymbol{\Lambda}^{-1}\boldsymbol{H}'(\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2}) \\
= \quad & (\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2})'(\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2})\boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_1^{-1/2}\boldsymbol{H}\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\Lambda}^{-1} \\
& (\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2}) \\
= \quad & (\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2})'\boldsymbol{\Lambda}^{-1}(\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2}).
\end{aligned} \tag{A.2}$$

After replacing Eq. (A.2) into Eq. (A.1), we have

$$\boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_0^{-1/2} = (\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2})'\boldsymbol{\Lambda}^{-1}(\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2}). \tag{A.3}$$

By comparing Eq. (4.13) and Eq. (A.3), we see that

$$\tilde{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}^{-1} \tag{A.4}$$

$$\tilde{\boldsymbol{H}} = (\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}'\boldsymbol{\Sigma}_1^{1/2}\boldsymbol{\Sigma}_0^{-1/2})'. \tag{A.5}$$

To find the relationship between $\tilde{\boldsymbol{\mu}}$ and $\boldsymbol{\mu}$, we start from Eq. (4.13):

$$\begin{aligned}
& \boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_0^{-1/2} = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{H}}' \\
\Longrightarrow \quad & \tilde{\boldsymbol{H}}'\boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_0^{-1/2} = \boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{H}}' \\
\Longrightarrow \quad & \tilde{\boldsymbol{H}}'\boldsymbol{\Sigma}_0^{-1/2}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_0^{-1/2}(\boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_1^{-1}) = \boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{H}}'(\boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_1^{-1}) \\
\Longrightarrow \quad & \tilde{\boldsymbol{H}}'\boldsymbol{\Sigma}_0^{-1/2} = \boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{H}}'\boldsymbol{\Sigma}_0^{1/2}\boldsymbol{\Sigma}_1^{-1}.
\end{aligned} \tag{A.6}$$

Replace Eqs. (A.5) and (A.6) into Eq. (4.14), we have

$$\begin{aligned}
\tilde{\boldsymbol{\mu}} &= \tilde{\boldsymbol{H}}'\boldsymbol{\Sigma}_0^{-1/2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \\
&= \boldsymbol{\Lambda}^{-1/2}\boldsymbol{H}'\boldsymbol{\Sigma}_1^{-1/2}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \\
&= -\boldsymbol{\Lambda}^{-1/2}\boldsymbol{\mu}.
\end{aligned} \tag{A.7}$$

APPENDIX B

MEAN AND VARIANCE OF THE DISCRIMINANT FUNCTION

To find the mean and variance of $Q^0_{d,n}$, we first replace Eqs. (4.8)-(4.11) into Eq. (4.12), and expand it to obtain

$$\begin{aligned} Q^0_{d,n} \sim & \frac{n-1}{2}\sum_{j=1}^{d}\left[\frac{n\lambda_j}{n+1}\frac{Z^2_{1j}}{T_2}+\frac{n+n\lambda_j+1}{n(n+1)}\frac{Z^2_{2j}}{T_2}+\frac{\mu_j^2}{T_2}\right. \\ & +2\frac{(\lambda_j(n+n\lambda_j+1))^{1/2}}{n+1}\frac{Z_{1j}Z_{2j}}{T_2}+2\left(\frac{n\lambda_j}{n+1}\right)^{1/2}\mu_j\frac{Z_{1j}}{T_2} \\ & \left.+2\left(\frac{n+n\lambda_j+1}{n(n+1)}\right)^{1/2}\mu_j\frac{Z_{2j}}{T_2}-\frac{n+1}{n}\frac{Z^2_{1j}}{T_1}\right] \\ & +\frac{1}{2}\left[\log\left(\frac{T_2}{T_1}\right)-\log|\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0|+\sum_{j=1}^{d-1}\log F_j\right]. \end{aligned} \tag{B.1}$$

It is easily shown that

$$\boldsymbol{E}\left[Z_{gj}\right]=0 \qquad g=1,2; j=1,2,\ldots,d \tag{B.2}$$

$$\boldsymbol{E}\left[Z^2_{gj}\right]=1 \qquad g=1,2; j=1,2,\ldots,d \tag{B.3}$$

$$\boldsymbol{E}\left[\frac{1}{T_l}\right]=\frac{1}{n-d-2} \qquad l=1,2 \tag{B.4}$$

$$\boldsymbol{E}\left[\log\left(\frac{T_2}{T_1}\right)\right]=0 \tag{B.5}$$

$$\boldsymbol{E}\left[\log F_j\right]=0 \qquad j=1,2,\ldots,d-1. \tag{B.6}$$

Taking the expectation of Eq. (B.1) with the help of Eqs. (B.2)-(B.6), we then have

$$\begin{aligned}\boldsymbol{E}\left[Q^0_{d,n}\right] &= \frac{n-1}{2}\sum_{j=1}^{d}\left(\frac{n\lambda_j}{n+1}\frac{1}{n-d-2}+\frac{n+n\lambda_j+1}{n(n+1)}\frac{1}{n-d-2}+\frac{\mu_j^2}{n-d-2}\right.\\ &\quad\left.-\frac{n+1}{n}\frac{1}{n-d-2}\right)-\frac{1}{2}\log|\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0|\\ &= \frac{1}{2}\frac{n-1}{n-d-2}\sum_{j=1}^{d}(\lambda_j+\mu_j^2-1)-\frac{1}{2}\sum_{j=1}^{d}\log\lambda_j \qquad \text{(B.7)}\end{aligned}$$

The calculation of the variance of $Q^0_{d,n}$ is a bit complicated. To make the whole procedure clear, we denote Eq. (B.1) as

$$\begin{aligned}Q^0_{d,n} &\sim A+B\\ &\sim \frac{n-1}{2}\sum_{j=1}^{d}(a_{1j}+a_{2j}+a_{3j}+a_{4j}+a_{5j}+a_{6j}-a_{7j})+\frac{1}{2}(b_1-b_2+b_3). \qquad \text{(B.8)}\end{aligned}$$

where

$$\begin{aligned}A =& \frac{n-1}{2}\sum_{j=1}^{d}(a_{1j}+a_{2j}+a_{3j}+a_{4j}+a_{5j}+a_{6j}-a_{7j})\\ B =& \frac{1}{2}(b_1-b_2+b_3)\end{aligned}$$

and

$$\begin{aligned}a_{1j} &= \frac{n\lambda_j}{n+1}\frac{Z_{1j}^2}{T_2} & a_{2j} &= \frac{n+n\lambda_j+1}{n(n+1)}\frac{Z_{2j}^2}{T_2}\\ a_{3j} &= \frac{\mu_j^2}{T_2} & a_{4j} &= 2\frac{(\lambda_j(n+n\lambda_j+1))^{1/2}}{n+1}\frac{Z_{1j}Z_{2j}}{T_2}\\ a_{5j} &= 2\left(\frac{n\lambda_j}{n+1}\right)^{1/2}\mu_j\frac{Z_{1j}}{T_2} & a_{6j} &= 2\left(\frac{n+n\lambda_j+1}{n(n+1)}\right)^{1/2}\mu_j\frac{Z_{2j}}{T_2}\\ a_{7j} &= \frac{n+1}{n}\frac{Z_{1j}^2}{T_1} & b_1 &= \log\left(\frac{T_2}{T_1}\right)\\ b_2 &= \log|\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_0| & b_3 &= \sum_{j=1}^{d-1}\log F_j.\end{aligned}$$

Then

$$\begin{aligned}\sigma^2_{Q^0_{d,n}} &= \boldsymbol{E}\left[(A+B-\boldsymbol{E}\left[A+B\right])^2\right] \\ &= \boldsymbol{var}\left[A\right]+\boldsymbol{var}\left[B\right]+2\left(\boldsymbol{E}\left[AB\right]-\boldsymbol{E}\left[A\right]\boldsymbol{E}\left[B\right]\right) \qquad \text{(B.9)}\end{aligned}$$

We now compute the three terms in the right hand side of Eq. (B.9) one by one.

1. $\boldsymbol{var}\left[A\right]$

Since

$$A=\frac{n-1}{2}\sum_{j=1}^{d}\left(a_{1j}+a_{2j}+a_{3j}+a_{4j}+a_{5j}+a_{6j}-a_{7j}\right),$$

the computation of $\boldsymbol{var}\left[A\right]$ naturally involves the variances of a_{kj}, $k=1,2,\ldots,7; j=1,2,\ldots,d$ and their cross-over terms. To calculate these terms, we need

$$\begin{aligned}&\boldsymbol{E}\left[Z^4_{gj}\right]=3 && g=1,2; j=1,2,\ldots,d \\ &\boldsymbol{E}\left[\frac{1}{T^2_l}\right]=\frac{1}{(n-d-2)(n-d-4)} && l=1,2.\end{aligned}$$

plus Eqs. (B.2)-(B.6) to obtain

$$\boldsymbol{var}\left[\frac{1}{T_l}\right]=\frac{2}{(n-d-2)^2(n-d-4)} \qquad l=1,2 \qquad \text{(B.10)}$$

$$\boldsymbol{var}\left[\frac{Z_{gj}}{T_l}\right]=\frac{1}{(n-d-2)(n-d-4)} \qquad g=1,2; l=1,2; j=1,2,\ldots,d \qquad \text{(B.11)}$$

$$\boldsymbol{var}\left[\frac{Z^2_{gj}}{T_l}\right]=\frac{2(n-d-1)}{(n-d-2)^2(n-d-4)} \qquad g=1,2; l=1,2; j=1,2,\ldots,d \qquad \text{(B.12)}$$

$$\boldsymbol{var}\left[\frac{Z_{1j}Z_{2j}}{T_l}\right]=\frac{1}{(n-d-2)(n-d-4)} \qquad l=1,2; j=1,2,\ldots,d, \qquad \text{(B.13)}$$

and

$$\boldsymbol{E}\left[\frac{Z_{gj}^2}{T_l^2}\right]-\boldsymbol{E}\left[\frac{Z_{gj}^2}{T_l}\right]\boldsymbol{E}\left[\frac{1}{T_l}\right]=\frac{2}{(n-d-2)^2(n-d-4)}$$
$$g=1,2;l=1,2;j=1,2,\ldots,d \tag{B.14}$$
$$\boldsymbol{E}\left[\frac{Z_{gj}^4}{T_1T_2}\right]-\boldsymbol{E}\left[\frac{Z_{gj}^2}{T_1}\right]\boldsymbol{E}\left[\frac{Z_{gj}^2}{T_2}\right]=\frac{2}{(n-d-2)^2}$$
$$g=1,2;j=1,2,\ldots,d \tag{B.15}$$
$$\boldsymbol{E}\left[\frac{Z_{gi}^2Z_{hj}^2}{T_l^2}\right]-\boldsymbol{E}\left[\frac{Z_{gi}^2}{T_l}\right]\boldsymbol{E}\left[\frac{Z_{hj}^2}{T_l}\right]=\frac{2}{(n-d-2)^2(n-d-4)}$$
$$g=1,2;h=1,2;l=1,2;i,j=1,2,\ldots,d;i\neq j \text{ or } g\neq h. \tag{B.16}$$

Then all the terms in $\boldsymbol{var}[A]$ can be computed according to Eqs. (B.2)-(B.6) and Eqs. (B.10)-(B.16). Table IX shows which equation in Eqs. (B.10)-(B.16) is used to calculate which term in computing $\boldsymbol{var}[A]$. By summing up all the terms, we have

$$\boldsymbol{var}[A]=\frac{(n-1)^2}{(n-d-2)^2(n-d-4)}\sum_{i=1}^{6}v_i \tag{B.17}$$

2. $\boldsymbol{var}[B]$

Since T_1, T_2, and F_j, $j=1,2,\ldots,d-1$, are mutually independent, and b_2 is a constant,

$$\begin{aligned}\boldsymbol{var}[B]&=\frac{1}{4}\left(\boldsymbol{var}[b_1]+\boldsymbol{var}[b_3]\right)\\&=\frac{1}{4}\left(\boldsymbol{var}[\log T_1]+\boldsymbol{var}[\log T_2]+\sum_{j=1}^{d-1}\boldsymbol{var}[F_j]\right).\end{aligned} \tag{B.18}$$

Since T_1 and T_2 are chi-square distribution with $n-d$ degree of freedom, we have

$$\boldsymbol{var}[\log T_1]=\boldsymbol{var}[\log T_2]=\psi'\left(\frac{n-d}{2}\right).$$

Note that F_j is F-distributed with $(n-j,n-j)$ degrees of freedom, thus F_j can

Table IX. The equations used in calculating the variance of a_{ij}, $i = 1, 2, \ldots, 7$, $j = 1, 2, \ldots, d$ and their cross-over terms. The upper-right triangle shows the terms among $a_{1j}, a_{2j}, \ldots, a_{7j}$, $j = 1, 2, \ldots, d$. The lower-left triangle shows the terms between $a_{1i}, a_{2i}, \ldots, a_{7i}$ and $a_{1j}, a_{2j}, \ldots, a_{7j}$, $i, j = 1, 2, \ldots, d, i \neq j$.

		a_{1j}	a_{2j}	a_{3j}	a_{4j}	a_{5j}	a_{6j}	a_{7j}	
		(B.12)	(B.16)	(B.14)	0	0	0	(B.15)	a_{1j}
a_{1i}	(B.16)		(B.12)	(B.14)	0	0	0	0	a_{2j}
a_{2i}	(B.16)	(B.16)		(B.10)	0	0	0	0	a_{3j}
a_{3i}	(B.14)	(B.14)	(B.10)		(B.13)	0	0	0	a_{4j}
a_{4i}	0	0	0	0		(B.11)	0	0	a_{5j}
a_{5i}	0	0	0	0	0		(B.11)	0	a_{6j}
a_{6i}	0	0	0	0	0	0		(B.12)	a_{7j}
a_{7i}	0	0	0	0	0	0	(B.16)		
	a_{1j}	a_{2j}	a_{3j}	a_{4j}	a_{5j}	a_{6j}	a_{7j}		

be viewed as the ratio between two independent chi-square distributions with $n - j$ degrees of freedom. Hence, similarly,

$$\boldsymbol{var}[\log F_j] = 2\psi'\left(\frac{n-j}{2}\right).$$

Thus

$$\begin{aligned}\boldsymbol{var}\,[B] &= \frac{1}{4}\left(2\psi'\left(\frac{n-d}{2}\right) + 2\sum_{j=1}^{d-1}\psi'\left(\frac{n-j}{2}\right)\right) \\ &= \frac{1}{2}\sum_{j=1}^{d}\psi'\left(\frac{n-j}{2}\right) \qquad \text{(B.19)}\end{aligned}$$

3. $\boldsymbol{E}\,[AB] - \boldsymbol{E}\,[A]\,\boldsymbol{E}\,[B]$

$$
\begin{aligned}
&2\left(\boldsymbol{E}\left[AB\right]-\boldsymbol{E}\left[A\right]\boldsymbol{E}\left[B\right]\right)\\
=&(\boldsymbol{E}\left[Ab_1\right]-\boldsymbol{E}\left[A\right]b_2+\boldsymbol{E}\left[A\right]\boldsymbol{E}\left[b_3\right])-(\boldsymbol{E}\left[A\right]\boldsymbol{E}\left[b_1\right]-\boldsymbol{E}\left[A\right]b_2+\boldsymbol{E}\left[A\right]\boldsymbol{E}\left[b_3\right])\\
=&\boldsymbol{E}\left[Ab_1\right]-\boldsymbol{E}\left[A\right]\boldsymbol{E}\left[b_1\right]\\
=&\boldsymbol{E}\left[Ab_1\right]\\
=&\boldsymbol{E}\left[(\log T_2-\log T_1)\frac{n-1}{2}\sum_{j=1}^{d}(a_{1j}+a_{2j}+a_{3j}+a_{4j}+a_{5j}+a_{6j}-a_{7j})\right]\\
=&\frac{n-1}{2}\left(\boldsymbol{E}\left[\log T_2\sum_{j=1}^{d}\sum_{i=1}^{6}a_{ij}\right]-\boldsymbol{E}\left[\log T_1\sum_{j=1}^{d}\sum_{i=1}^{6}a_{ij}\right]\right.\\
&\left.-\boldsymbol{E}\left[\log T_2\sum_{j=1}^{d}a_{7j}\right]+\boldsymbol{E}\left[\log T_1\sum_{j=1}^{d}a_{7j}\right]\right)
\end{aligned}
\tag{B.20}
$$

where the first equation comes from the fact that b_2 is a constant and $b_3=\sum_{j=1}^{d-1}\log F_j$ is independent of A, and the third equation comes from Eq. (B.5).

By using Eqs. (B.2)-(B.6), we then have

$$
\begin{aligned}
\boldsymbol{E}\left[\log T_2\sum_{j=1}^{d}\sum_{i=1}^{6}a_{ij}\right]=&\boldsymbol{E}\left[\frac{\log T_2}{T_2}\right]\sum_{j=1}^{d}\left(\frac{n\lambda_j}{n+1}+\frac{n+n\lambda_j+1}{n(n+1)}+\mu_j^2\right)\\
=&\boldsymbol{E}\left[\frac{\log T_2}{T_2}\right]\sum_{j=1}^{d}\left(\lambda_j+\mu_j^2+\frac{1}{n}\right),
\end{aligned}
\tag{B.21}
$$

$$
\begin{aligned}
\boldsymbol{E}\left[\log T_1\sum_{j=1}^{d}\sum_{i=1}^{6}a_{ij}\right]=&\boldsymbol{E}\left[\log T_1\right]\sum_{j=1}^{d}\left(\frac{n\lambda_j}{(n+1)(n-d-2)}\right.\\
&\left.+\frac{n+n\lambda_j+1}{n(n+1)(n-d-2)}+\frac{\mu_j^2}{n-d-2}\right)\\
=&\frac{\boldsymbol{E}\left[\log T_1\right]}{n-d-2}\sum_{j=1}^{d}\left(\lambda_j+\mu_j^2+\frac{1}{n}\right),
\end{aligned}
\tag{B.22}
$$

$$\boldsymbol{E}\left[\log T_2 \sum_{j=1}^{d} a_{7j}\right] = \frac{\boldsymbol{E}\left[\log T_2\right]}{n-d-2}\frac{n+1}{n}d, \tag{B.23}$$

and

$$\boldsymbol{E}\left[\log T_1 \sum_{j=1}^{d} a_{7j}\right] = \boldsymbol{E}\left[\frac{\log T_1}{T_1}\right]\frac{n+1}{n}d. \tag{B.24}$$

Since T_1 and T_2 are chi-square distributions with $n-d$ degree of freedom, we have

$$\boldsymbol{E}\left[\log T_1\right] = \boldsymbol{E}\left[\log T_2\right] = \frac{\log 2 + \psi(\frac{n-d}{2})}{n-d-2} \tag{B.25}$$

$$\boldsymbol{E}\left[\frac{\log T_1}{T_1}\right] = \boldsymbol{E}\left[\frac{\log T_2}{T_2}\right] = \frac{\log 2 + \psi(\frac{n-d}{2})}{n-d-2} - \frac{2}{(n-d-2)^2} \tag{B.26}$$

Replacing Eqs. (B.21)-(B.25) into Eq. (B.20), we can merge the terms into

$$\begin{aligned} 2\left(\boldsymbol{E}\left[AB\right] - \boldsymbol{E}\left[A\right]\boldsymbol{E}\left[B\right]\right) &= \left(\boldsymbol{E}\left[\frac{\log T_1}{T_1}\right] - \frac{\boldsymbol{E}\left[\log T_1\right]}{n-d-2}\right)\frac{n-1}{2}\sum_{i=1}^{d}\left(\lambda_j + \mu_j^2 + \frac{n+2}{n}\right) \\ &= -\frac{n-1}{(n-d-2)^2}\sum_{i=1}^{d}\left(\lambda_j + \mu_j^2 + \frac{n+2}{n}\right) \\ &= -\frac{(n-1)^2}{(n-d-2)^2(n-d-4)}c \end{aligned} \tag{B.27}$$

VITA

Jianping Hua received the B.E. degree and M.S. degree in electronic engineering from Tsinghua University, P.R. China, in 1998 and 2000, respectively. He has been pursing the Ph.D. degree in electrical engineering as Texas A&M University since 2000. Hua has worked as a research assistant in the Multimedia Lab since 2000, and the Genomic Signal Processing Lab since 2003, both in the Department of Electrical Engineering, Texas A&M University. He was a research intern at the Advanced Digital Imaging Research, LLC, League City, TX, in the summer of 2001.

Hua can be reached at the following address

P.O. Box 75
China Agricultural University (East Campus)
Haidian District
Beijing, 100083
P.R. China

The typist for this thesis was Jianping Hua.